Pose Tracking of Multiple Camera System

LEUNG, Man Kin

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Master of Philosophy
in
Computer Science and Engineering

October 2008

Thesis/Assessment Committee

Professor JIA, Jiaya Leo (Chair)

Professor WONG, Kin Hong (Thesis Supervisor)

Professor MOON, Yiu Sang (Committee Member)

Professor TSANG, Wai Ming Peter (External Examiner)

Abstract of thesis entitled:

Pose Tracking of Multiple Camera System

Submitted by LEUNG, Man Kin

for the degree of Master of Philosophy

at The Chinese University of Hong Kong in October 2008

Recovering the orientation and location of a camera from image sequences is an important and challenging problem in computer vision. There are many related applications. For example, it can help to create augmented reality in films, in particular, artificial objects can be inserted into the video easily after knowing the orientation and location of the camera and the 3-D model of the scene. Also, cameras can be mounted on mobile objects like robots to track the motions of the objects.

In this thesis, we focus on pose tracking of multiple camera system in a model-less scheme which means that the motion of the system is estimated without both the prior knowledge and explicit calculation of the structure. Firstly, we apply different estimation methods based on this model-less scheme to recover the orientation and location of a stereo camera system. Throughout the experiment, advantages and disadvantages of different estimation methods are compared and analyzed. We also find that our approach improves over the existing algorithms in terms of accuracy.

We then propose an algorithm to track the pose of a multiple camera system that consists of two pairs of stereo cameras. One advantage of using multiple stereo cameras is to increase the field of view to capture more information for computing the pose. Therefore, our approach can still work even if either one of the stereo pairs is occluded. The proposed algorithm has another advantage that neither the prior knowledge nor computation of the 3-D structure of the scene is required. The orientation and location of the system are recovered efficiently and directly by employing the trifocal tensor constraints. Experiments show that our approach is more accurate than the previous approaches.

We believe that our approaches are useful for applications related to camera pose tracking like building augmented reality system, navigating robot, and sensing motion.

從連續影像計算攝像機的方向與位置在計算機視覺中是一個重要與富挑戰性的題目。有很多實際應用與這個題目有關。例如：這可以幫助電影製作擴增實境。當知道攝像機的方向與位置和場境的三維模型後，我們可以將虛擬物件插入電影中。另外，我們可以安裝攝像機在一些可動物件如機器人上去追蹤可動物件的移動。

在這篇論文中，我們集中探討以無模型方案去追蹤多攝像機系統的移動。無模型方案是指在估計多攝像機系統的動作中，不需要知道或計算模型的結構。首先，我們以這個無模型方案去運用不同的估計方法去追蹤一對立體攝像機的方向與位置。經過一連串的實驗，我們比較了不同估計方法的好處與壞處。我們並發現我們的方法比現在的方法更爲準確。

我們跟著再建議一個方法去追蹤一個由兩對立體攝像機組成的多攝像機系統的方向與位置。其中一個好處是可以增加可視範圍以獲得更多數據去計算這個多攝像機系統的動作。所以，當其中一對立體攝像機受遮擋時，我們的方法仍可以繼續運行。我們的方法的另一個好處是不需要知道或計算模型的結構。我們使用三焦距張量的限制有效率地直接計算出多攝像機系統的方向與位置。實驗證明，我們的方法比之前的方法更加準確。

我們相信我們的方法對於製作擴增實境系統、機械人導航和感應動作等應用都有所幫助。

Acknowledgement

First of all, I would like to give wholehearted thanks to my supervisor, Prof. Kin Hong Wong, who has given me a lot of support, guidance, and advices throughout the past two years.

I would also like to sincere thanks to my fellow research partner, Eric Yu, who always kindly gives me helping hand to my research.

Moreover, I would like to thanks Prof. Leo Jiaya Jia and Prof. Yiu Sang Moon for being my internal examiners.

Last but not least, I want to give thanks to my dear friends, Wyman Wong, Albert Lam, Oscar Leung, KK Lo, Edith Ngai, Pat Chan, Alan Chu, Brian Tsui, Jim, Matthew Tang, and many others. The days we study together are happy and full of joy. I really enjoy the time with them.

This work is dedicated to my family for their continuous encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Estimating the orientation and location of a camera from image sequences is an important problem in computer vision. There are many interesting applications related to this kind of research. For example, it can help creating augmented reality in films by inserting artificial objects into the footage easily when the orientation and location of the camera are known. It can also help robot navigating by mounting cameras on the robot. Unlike other mechanical sensors such as accelerometers, cameras can get the visual information of the surroundings to compute the orientation and location of themselves while accelerometers can only measure their own accelerations.

Camera pose estimation is a challenging research problem. Many methods were proposed to solve the problem. Their methods can be classified into two main approaches. One of them is marker-based approach. The scene must be specially designed to contain markers whose 3-D positions in the scene have al-

ready been inputted to the computers. As a result, the pose is estimated given the prior knowledge of the scene. Another one is vision-based approach in which no special control of the scene is required. Therefore, the approach is more useful as its applications do not need to be confined to a special environment. As there are no predefined markers in the scene, most research work detects and matches representative patches called features in the images and uses the 2-D coordinates of their positions to compute the pose.

The work in this thesis belongs to the vision-based approach. Given the 2-D positions of the features, we want to recover the orientation and translation of the camera. However, because of the relationships between the 2-D coordinates of the features' positions and the orientation and location of the camera, the 3-D coordinates of the features' positions in the scene corresponding to their projections (2-D coordinates) on the images are also computed in addition to the camera pose. Most traditional algorithms are based on this approach and this kind of research belongs to structure and motion. The name implies both the 3-D coordinates of the points in the scene (structure) and the camera pose (motion) are estimated simultaneously. Besides structure and motion, there is research on recovering the camera pose directly without both the prior knowledge and explicit computation of the structure. Our approach belongs to this category.

In the real world, there are no perfect cameras, and thus the images contain noise. The 2-D coordinates of the features' positions in the image may not be exact and the matched features

may be wrong. These two factors would affect the accuracy of the recovered camera pose. This is one of the major challenges in this field. To deal with this problem, early approaches used numerical methods like Newton's method to optimize the result by considering the information of all the images. These approaches can achieve high accuracy. However, the drawback is that it is time-consuming and belongs to batch processing, which means all the images in a sequence are used for the processing, and thus the processing time is too long for many applications.

Recently, some approaches were proposed to tackle the problem in real-time. These approaches allow the current camera pose to be recovered immediately just after the current image has been taken. Therefore, the domain of their applications increases. For example, it can help tracking the movement of a mobile object likes robot in real-time by mounting cameras on it. Although these approaches are efficient, the accuracy of these approaches is lower. It depends on several factors like feature management and estimation method. Feature management means how the features are handled. When a camera moves around, the view from the camera changes and thus appearing of new features and disappearing of original features occur. In addition, the features are not necessarily reliable as they may be mismatched among the images. As a result, some techniques are needed to handle these features. Estimation method means the method used to estimate the solution. Most approaches make use of some recursive filters like the Kalman filter to estimate the solution. Different estimation methods usually model the problem and noise using different ways.

Besides research on a single camera, there is research on pose tracking of a multiple camera system. It is important to decide the number of cameras and the position of each camera in the system as they would affect the data obtained from the cameras. Cameras with overlapping views like a stereo camera system can enable us to get reliable features and the depth of scene. On the other hand, cameras with non-overlapping views can have the advantages of larger field of view. As a result, it poses more challenges and potentials in this field.

1.2 Motivation

There is an interesting work [38] focusing on recovering the location and orientation of a pair of stereo cameras using the image sequences. It uses an extended Kalman filter to track the pose of the stereo camera system. In the filter, it makes use of an approximated twist motion model in the dynamic model and trifocal tensors as constraints to bypass the computation of the structure to estimate the camera pose.

The work leads to several interesting problems which should be faced by other similar approaches. Firstly, the projection of the 3-D point on the image is a non-linear function. However, the extended Kalman filter, used by many other similar approaches, handles a non-linear system by assuming local linearity of the system. As a result, it is valuable to investigate if the extended Kalman filter is a good choice and if there are other suitable methods for this kind of problems. Secondly, the motion model of the camera is crucial when we track the pose

from image sequences. Different motion models would result in different accuracies. Therefore, it is interesting to investigate the motion model of the camera.

In addition, using either a camera or a stereo camera system would face a problem. The pose of the system can not be tracked if the field of view does not contain enough information (reliable features). In this case, using more cameras with non-overlapping views can solve this problem. When a camera does not get enough features, cameras with other views can continue to help to track the pose of the whole system. In particular, we work on a multiple camera system consisting of two pairs of stereo cameras. Each stereo pair can get reliable features from different views to track the pose. As a result, the pose of the multiple camera system can be recovered even if one stereo pair cannot get enough features.

1.3 Contributions

- We propose an algorithm to track the pose of a stereo camera system.
  - Our algorithm can recover more accurate pose than the existing work by using the Rodrigues' formula, in which no approximations are taken.
  - Performances of different estimation methods including the extended Kalman filter, unscented Kalman filter, and differential evolution used in our approach are compared and analyzed. Advantages and disadvantages of

them are discussed.

- We propose an algorithm to track the pose of a multiple camera system consisting of two pairs of stereo cameras.
  - The proposed algorithm can recover the orientation and location of the multiple camera system accurately and efficiently.
  - Our algorithm can work even when one stereo pair is blocked.
  - Performances of different estimation methods including the extended Kalman filter and unscented Kalman filter used in our approach are compared and analyzed. Their advantages and disadvantages are discussed.
  - Different orientations between the two stereo pairs are studied to investigate their effects on the accuracy of the pose.

1.4 Organization of the thesis

Chapter 2 is the literature review. It briefly introduces the background knowledge and surveys the recent research work on structure and motion and camera localization.

Chapter 3 is a chapter focusing on the pose estimation of stereo cameras. It proposes an approach to recover the orientation and location of the stereo cameras. Different estimation methods including the unscented Kalman filter, extended

Kalman filter, and differential evolution are discussed and analyzed for the problem.

Chapter 4 focuses on the pose estimation of two pairs of stereo cameras. It proposes a new approach to recover the orientation and location of the multiple camera system consisting of two pairs of the stereo cameras. Different Kalman filters including the unscented Kalman filter and extended Kalman filter are applied and analyzed for the problem.

Chapter 5 concludes the work in the thesis.

□ **End of chapter.**

Chapter 2

Literature review

2.1 Introduction

In this chapter, background knowledge for this thesis is briefly described and the work related to the field of structure and motion is discussed. We classify the field into three main categories as shown in figure 2.1. Those categories are batch method, sequential method, and method solving the problem of simultaneous localization and mapping (SLAM) using cameras.

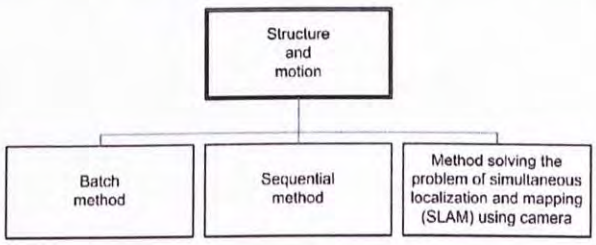

Figure 2.1: Classification of methods in the field of structure and motion (SAM).

Batch methods refer to the methods in which all the images must be ready before the processing. All the images are used when computing the camera pose and 3-D structure of the scene

using some numerical methods. Therefore, it is only suitable for off-line processing and thus the number of applications is limited.

Sequential methods do not require that all the images must be ready at the beginning of the process. This kind of methods estimates the camera pose right after the image has been taken. The approaches belonging to this kind usually involve the use of some recursive filters such as the Kalman filter and the particle filter. Because of their efficiencies, they are expected to be applicable in real-time.

SLAM originally does not belong to the field of computer vision, but the field of robotic system. The target of SLAM is to locate a system consisting of a number of different sensors and map the surrounding scene into a 3-D model simultaneously. However, some recent research tried to use cameras as the only kind of sensors. Therefore, the problem becomes a structure and motion problem. Most of the approaches made uses of the particle filter or the Kalman filter to tackle the problem in real-time.

In the followings, background knowledge of the proposed approaches in this thesis is described.

2.2 Background knowledge

The concepts of the pinhole camera model and the Kalman filter are frequently revisited in the following chapters. They are briefly introduced in the following sections.

2.2.1 Pinhole camera model

The pinhole camera model which is illustrated in figure 2.2 is used throughout the thesis. The relationship between the mth

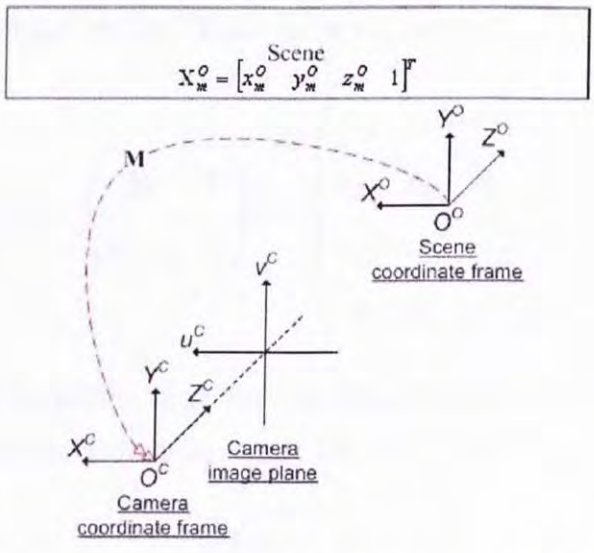

Figure 2.2: Image system of the pinhole camera model.

3-D model point $\mathbf{X}_m^O$ in the scene and its projection $\mathbf{u}_m$ on the image plane is

$$\mathbf{u}_m = \mathbf{K}\left(\mathbf{I}_{3\times3} \quad \mathbf{0}_{3\times1}\right)\mathbf{M}\mathbf{X}_m^O$$

$$\begin{pmatrix} \bar{u}_m \\ \bar{v}_m \\ \bar{w}_m \end{pmatrix} = \mathbf{K}\left(\mathbf{I}_{3\times3} \quad \mathbf{0}_{3\times1}\right)\mathbf{M}\begin{pmatrix} x_m^O \\ y_m^O \\ z_m^O \\ 1 \end{pmatrix} \tag{2.1}$$

$\mathbf{X}_m^O = [x_m^O \; y_m^O \; z_m^O \; 1]^T$ are the homogeneous coordinates of the mth 3-D model point $[x_m^O \; y_m^O \; z_m^O]^T$ in the object coordinate frame.

$\mathbf{M}$ is a 4×4 matrix that encapsulates the extrinsic parameters of the camera and transforms 3-D model points from the object coordinate frame to the camera coordinate frame. The transformation is consisting of a 3×3 rotation matrix $\mathbf{R}$ and a 3×1 translation vector $\mathbf{T}$ as shown in equation (2.2).

$$\mathbf{M} = \begin{pmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0}_{1\times3} & 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.2}$$

$\mathbf{K}$ is a 3×3 matrix that encapsulates the intrinsic parameters of the camera as shown in equation (2.3).

$$\mathbf{K} = \begin{pmatrix} fm_x & 0 & x_o \\ 0 & fm_y & y_o \\ 0 & 0 & 1 \end{pmatrix} \tag{2.3}$$

f is the focal length of the camera. $[x_o\ y_o]^T$ are the coordinates of the principal point. $[m_x\ m_y]^T$ represent the dimension of a pixel.

$[\bar{u}_m\ \bar{v}_m\ \bar{w}_m]^T$ are the homogeneous coordinates of the projection $[\frac{\bar{u}_m}{\bar{w}_m}\ \frac{\bar{v}_m}{\bar{w}_m}]^T$ of the mth 3-D point on the image plane of the camera.

2.2.2 Kalman filter

The Kalman filter [11] is a recursive filter that estimates the state of a dynamic system from a series of noisy measurements

and controls. The noises of the system are modeled by Gaussian random variables and handled by covariance matrices. To use the Kalman filter, the *state*, *dynamic model*, and *measurement model* of the dynamic system must be defined. Consider that the state is $\mathbf{x}_t$, the dynamic model is

$$\mathbf{x}_t = \mathbf{F}_t\mathbf{x}_{t-1} + \mathbf{B}_t\mathbf{u}_{t-1} + \mathbf{v} \tag{2.4}$$

and the measurement model is

$$\mathbf{y}_t = \mathbf{H}_t\mathbf{x}_t + \mathbf{n} \tag{2.5}$$

In equation (2.4), $\mathbf{F}_t$ is the state transition matrix which is applied to the state vector $\mathbf{x}_{t-1}$, $\mathbf{B}_t$ is the control-input matrix which is applied to the control vector $\mathbf{u}_{t-1}$, and $\mathbf{v}$ represents the process noise. In equation (2.5), $\mathbf{H}_t$ is the measurement matrix which is applied to the state vector $\mathbf{x}_t$ to obtain the measurements, and $\mathbf{n}$ represents the measurement noise.

Having the state, dynamic model, and measurement model defined, equations required for the Kalman filter can be derived. The procedure of the estimation of the state is divided into two stages illustrated in figure 2.3. The first one is the prediction (time update) stage. The current state, input control, and process noise are used to predict the next state and covariance. Another stage is the updating (measurement update) stage. The estimated state and measurements are used to update the next state and covariance. The corresponding time update equations

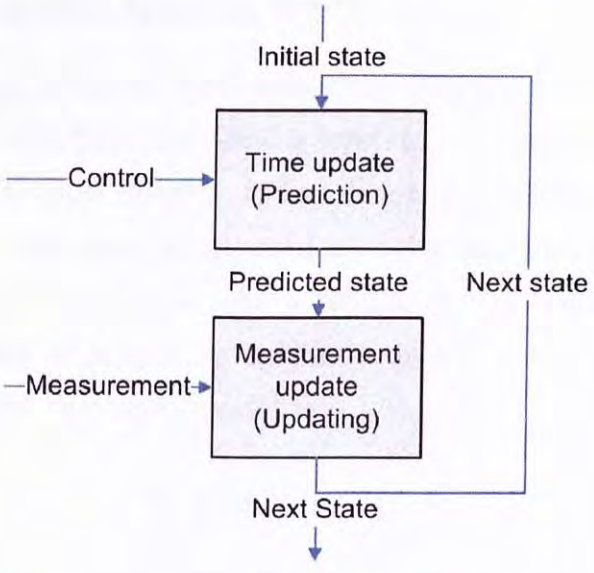

Figure 2.3: Illustration of the stages in the Kalman filter.

are

$$\begin{aligned} \hat{\mathbf{x}}_t^- &= \mathbf{F}_t\hat{\mathbf{x}}_{t-1} + \mathbf{B}_t\mathbf{u}_{t-1} \\ \mathbf{P}_t^- &= \mathbf{F}_t\mathbf{P}_{t-1}{\mathbf{F}_t}^T + \mathbf{R}^{\mathbf{v}} \end{aligned} \tag{2.6}$$

and the corresponding measurement update equations are

$$\begin{aligned} \mathbf{K}_t &= \mathbf{P}_t^-{\mathbf{H}_t}^T(\mathbf{H}_t\mathbf{P}_t^-{\mathbf{H}_t}^T + \mathbf{R}^{\mathbf{n}})^{-1} \\ \hat{\mathbf{x}}_t &= \hat{\mathbf{x}}_t^- + \mathbf{K}_t(\mathbf{y}_t - \mathbf{H}_t\hat{\mathbf{x}}_t^-) \\ \mathbf{P}_t &= (\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)\mathbf{P}_t^- \end{aligned} \tag{2.7}$$

$\hat{\mathbf{x}}_t^-$ and $\hat{\mathbf{x}}_t$ represent the states at time t after time update and measurement update respectively. $\mathbf{P}_t^-$ and $\mathbf{P}_t$ are the covariance matrices of the states $\hat{\mathbf{x}}_t^-$ and $\hat{\mathbf{x}}_t$ respectively. $\mathbf{R}^{\mathbf{v}}$ and $\mathbf{R}^{\mathbf{n}}$ are the covariance matrices of the process noise $\mathbf{v}$ and the measurement noise $\mathbf{n}$ respectively. $\mathbf{y}_t$ is the actual measurement. $\mathbf{K}_t$ is the **Kalman gain matrix in the Kalman filter.**

2.2.3 Extended Kalman filter

The ordinary Kalman filter can only handle a linear dynamic system. To handle a non-linear system, the extended Kalman filter [11] is a good option. It handles the non-linear system by linearizing the system locally and uses the Jacobian to propagate the state and covariance. As a result, the posterior state and covariance are accurate to the first order. Consider that the state is $\mathbf{x}_t$, the dynamic model is

$$\mathbf{x}_t = \mathbf{f}_t(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{v}) \tag{2.8}$$

and the measurement model is

$$\mathbf{y}_t = \mathbf{h}_t(\mathbf{x}_t, \mathbf{n}) \tag{2.9}$$

In equation (2.8), $\mathbf{f}_t$ is the state transition function which is applied to the state vector $\mathbf{x}_{t-1}$, control vector $\mathbf{u}_{t-1}$, and process noise $\mathbf{v}$ to have the next state. In equation (2.9), $\mathbf{h}_t$ is the measurement function which is applied to the state vector $\mathbf{x}_t$ and measurement noise $\mathbf{n}$ to get the measurements.

Same as the ordinary Kalman filter, it consists of prediction (time update) stage and updating (measurement update) stage. The corresponding time update equations are

$$\begin{aligned} \hat{\mathbf{x}}_t^- &= \mathbf{f}_t(\hat{\mathbf{x}}_{t-1}, \mathbf{u}_{t-1}, \mathbf{0}) \\ \mathbf{P}_t^- &= \mathbf{F}_t \mathbf{P}_{t-1} {\mathbf{F}_t}^T + \mathbf{V}_t \mathbf{R}^{\mathbf{v}} {\mathbf{V}_t}^T \end{aligned} \tag{2.10}$$

and the corresponding measurement update equations are

$$\begin{aligned} \mathbf{K}_t &= \mathbf{P}_t^- \mathbf{H}_t{}^T (\mathbf{H}_t \mathbf{P}_t^- \mathbf{H}_t{}^T + \mathbf{N}_t \mathbf{R}^{\mathbf{n}} \mathbf{N}_t{}^T)^{-1} \\ \hat{\mathbf{x}}_t &= \hat{\mathbf{x}}_t^- + \mathbf{K}_t(\mathbf{y}_t - \mathbf{h}_t(\hat{\mathbf{x}}_t^-, 0)) \\ \mathbf{P}_t &= (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t)\mathbf{P}_t^- \end{aligned} \tag{2.11}$$

$\hat{\mathbf{x}}_t^-$ and $\hat{\mathbf{x}}_t$ represent the states at time t after time update and measurement update respectively. $\mathbf{P}_t^-$ and $\mathbf{P}_t$ are the covariance matrices of the state $\hat{\mathbf{x}}_t^-$ and $\hat{\mathbf{x}}_t$ respectively. $\mathbf{R}^{\mathbf{v}}$ and $\mathbf{R}^{\mathbf{n}}$ are the covariance matrices of the process noise $\mathbf{v}$ and the measurement noise $\mathbf{n}$ respectively. $\mathbf{y}_t$ is the actual measurement. $\mathbf{F}_t$ is the Jacobian matrix of partial derivatives of $\mathbf{f}_t$ with respect to $\mathbf{x}$. $\mathbf{V}_t$ is the Jacobian matrix of partial derivatives of $\mathbf{f}_t$ with respect to $\mathbf{v}$. $\mathbf{H}_t$ is the Jacobian matrix of partial derivatives of $\mathbf{h}_t$ with respect to $\mathbf{x}$. $\mathbf{N}_t$ is the Jacobian matrix of partial derivatives of $\mathbf{h}_t$ with respect to $\mathbf{n}$. $\mathbf{K}_t$ is the Kalman gain matrix in the extended Kalman filter.

2.2.4 Unscented Kalman filter

Besides the extended Kalman filter, the unscented Kalman filter [18] [16] is an alternative which is able to handle a non-linear system. Instead of assuming local linearity, the unscented transform is used to propagate the state and covariance of the dynamic system by statistical calculations. The unscented transform is a method for computing the statistics of a random variable undergoing a non-linear transformation. In the unscented transform, a minimal set of sample points called sigma points completely capturing the true mean and covariance of the Gaus-

sian random variable are chosen to propagate through the true non-linear system. After the propagating, the posterior mean and covariance can be accurate to the second order. Therefore, the unscented Kalman filter can handle a non-linear system better than the extended Kalman filter which is accurate to the first order. Consider that we have the same state, dynamic model and measurement model as described in section 2.2.3, the required equations can be derived.

Unlike the extended Kalman filter, the unscented Kalman filter needs an initialization for the state and covariance to prepare the generation of the sigma points. The initialization is

$$\begin{aligned} \hat{\mathbf{x}}_0^a &= \begin{pmatrix} \hat{\mathbf{x}}_0^T & \mathbf{0} & \mathbf{0} \end{pmatrix}^T \\ \mathbf{P}_0^a &= \begin{pmatrix} \mathbf{P}_0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^{\mathbf{v}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}^{\mathbf{n}} \end{pmatrix} \end{aligned} \tag{2.12}$$

The sigma points used in the unscented transform are generated by

$$\mathbf{X}_{t-1}^a = \begin{bmatrix} \hat{\mathbf{x}}_{t-1}^a & \hat{\mathbf{x}}_{t-1}^a + \sqrt{(L+\lambda)\mathbf{P}_{t-1}^a} & \hat{\mathbf{x}}_{t-1}^a - \sqrt{(L+\lambda)\mathbf{P}_{t-1}^a} \end{bmatrix} \tag{2.13}$$

at each time step.

The corresponding time update equations are

$$\begin{aligned}
\mathbf{X}^x_{t|t-1} &= \mathbf{F}^*_t(\mathbf{X}^x_{t-1}, \mathbf{u}_{k-1}, \mathbf{X}^v_{t-1}) \\
\hat{\mathbf{x}}^-_t &= \sum_{i=0}^{2L} W_i^{(m)} X^x_{i,t|t-1} \\
\mathbf{P}^-_t &= \sum_{i=0}^{2L} W_i^{(c)} (X^x_{i,t|t-1} - \hat{\mathbf{x}}^-_t)(X^x_{i,t|t-1} - \hat{\mathbf{x}}^-_t)^T \\
\mathbf{Y}_{t|t-1} &= \mathbf{H}^*_t(\mathbf{X}^x_{t|t-1}, \mathbf{X}^n_{t|t-1}) \\
\hat{\mathbf{y}}^-_t &= \sum_{i=0}^{2L} W_i^{(m)} Y_{i,t|t-1}
\end{aligned} \tag{2.14}$$

and the corresponding measurement update equations are

$$\begin{aligned}
\mathbf{P}_{\tilde{\mathbf{y}}_t\tilde{\mathbf{y}}_t} &= \sum_{i=0}^{2L} W_i^{(c)} (Y_{i,t|t-1} - \hat{\mathbf{y}}^-_t)(Y_{i,t|t-1} - \hat{\mathbf{y}}^-_t)^T \\
\mathbf{P}_{\mathbf{x}_t\mathbf{y}_t} &= \sum_{i=0}^{2L} W_i^{(c)} (X_{i,t|t-1} - \hat{\mathbf{x}}^-_t)(Y_{i,t|t-1} - \hat{\mathbf{y}}^-_t)^T \\
\mathbf{K}_t &= \mathbf{P}_{\mathbf{x}_t\mathbf{y}_t} \mathbf{P}_{\tilde{\mathbf{y}}_t\tilde{\mathbf{y}}_t}{}^{-1} \\
\hat{\mathbf{x}}_t &= \hat{\mathbf{x}}^-_t + \mathbf{K}_t(\mathbf{y}_t - \hat{\mathbf{y}}^-_t) \\
\mathbf{P}_t &= \mathbf{P}^-_t - \mathbf{K}_t \mathbf{P}_{\tilde{\mathbf{y}}_t\tilde{\mathbf{y}}_t} \mathbf{K}_t{}^T
\end{aligned} \tag{2.15}$$

where

$$\begin{aligned}
\mathbf{x}^a &= \begin{pmatrix} \mathbf{x}^T & \mathbf{v}^T & \mathbf{n}^T \end{pmatrix}^T \\
\mathbf{X}^a &= \begin{pmatrix} (\mathbf{X}^x)^T & (\mathbf{X}^v)^T & (\mathbf{X}^n)^T \end{pmatrix}^T \\
W_0^{(m)} &= \frac{\lambda}{L+\lambda} \\
W_0^{(c)} &= \frac{\lambda}{L+\lambda} + 1 - \alpha^2 + \beta \\
W_i^{(m)} &= W_i^{(c)} = \frac{1}{2(L+\lambda)}, \; i = 1, 2, ..., 2L
\end{aligned} \tag{2.16}$$

$\hat{\mathbf{x}}_t^-$ and $\hat{\mathbf{x}}_t$ represent the states at time t after time update and measurement update respectively. $\mathbf{P}_t^-$ and $\mathbf{P}_t$ are the covariance matrices of the state $\hat{\mathbf{x}}_t^-$ and $\hat{\mathbf{x}}_t$ respectively. $\mathbf{R^v}$ and $\mathbf{R^n}$ are the covariance matrices of the process noise $\mathbf{v}$ and the measurement noise $\mathbf{n}$ respectively. $\mathbf{y}_t$ is the actual measurement. $\mathbf{K}_t$ is the Kalman gain matrix in the unscented Kalman filter. λ is a scaling parameter. α and β represent the spread of the sigma points and the prior knowledge of the distribution of the state $\mathbf{x}_t$ respectively. $\mathbf{X}_{t-1}^a$ contains all the sigma points used in the unscented transform while $\mathbf{X}_{i,t-1}^a$ indicates the ith sigma point. L is the dimension of the state $\hat{\mathbf{x}}_{t-1}^a$. $W_0^{(m)}$ and $W_0^{(c)}$ are the weights used in calculating the mean and the covariance matrices respectively. $\mathbf{F}_t^*(\mathbf{X}_{t-1}^x, \mathbf{u}_{k-1}, \mathbf{X}_{t-1}^v)$ is a function which calculates all the sigma points using $\mathbf{f}_t(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{v})$ defined in equation

(2.8). That is

$$\begin{aligned} &\mathbf{X}^x_{t|t-1} = \mathbf{F}^*_t(\mathbf{X}^x_{t-1}, \mathbf{u}_{k-1}, \mathbf{X}^v_{t-1}) \\ &\text{means} \\ &\mathbf{X}^x_{i,t|t-1} = \mathbf{f}_t(\mathbf{X}^x_{i,t-1}, \mathbf{u}_{k-1}, \mathbf{X}^v_{i,t-1}) \\ &i \in (1, 2, ..., 2L, 2L+1) \end{aligned} \tag{2.17}$$

$\mathbf{H}^*_t(\mathbf{X}^x_{t|t-1}, \mathbf{X}^n_{t|t-1})$ is a function which calculates all the sigma points using $\mathbf{h}_t(\mathbf{x}_t, \mathbf{n})$ defined in equation (2.9). That is

$$\begin{aligned} &\mathbf{Y}_{t|t-1} = \mathbf{H}^*_t(\mathbf{X}^x_{t|t-1}, \mathbf{X}^n_{t|t-1}) \\ &\text{means} \\ &\mathbf{Y}_{i,t|t-1} = \mathbf{h}_t(\mathbf{X}^x_{i,t|t-1}, \mathbf{X}^n_{i,t|t-1}) \\ &i \in (1, 2, ..., 2L, 2L+1) \end{aligned} \tag{2.18}$$

Details of the unscented Kalman filter and unscented transform can be found in [34].

2.3 Batch method

In this section, different batch methods which require all the images have to be ready before the processing are introduced. They include multiple view geometry [13], factorization [31], and bundle adjustment [33].

2.3.1 Multiple view geometry

There was much research focusing on the multiple view geometry [13]. The most basic type is two-view geometry. Consider there

are two image points $\mathbf{u}^{C1}$ and $\mathbf{u}^{C2}$, in two images respectively, representing the same point in the scene as illustrated in figure 2.4, there is a relationship between the two points and a 3×3 matrix called fundamental matrix $\mathbf{F}$ as shown in equation (2.19).

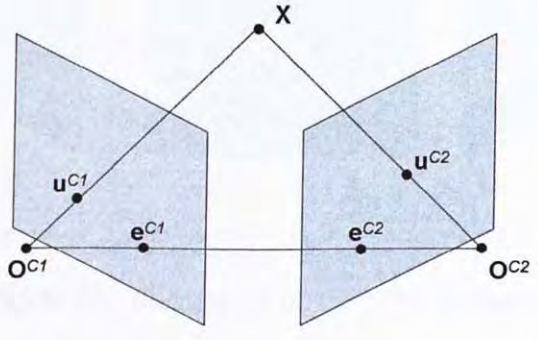

Figure 2.4: Illustration of the fundamental matrix.

$$\mathbf{u}^{C2^T}\mathbf{F}\mathbf{u}^{C1} = \mathbf{0} \tag{2.19}$$

To calculate the fundamental matrix $\mathbf{F}$, 8 corresponding points in two views are required.

If the camera is calibrated with the matrix $\mathbf{K}$ representing the intrinsic parameters as defined in equation (2.3), a 3×3 matrix called essential matrix $\mathbf{E}$ can be found using equation (2.20).

$$\mathbf{E} = \mathbf{K}^T\mathbf{F}\mathbf{K} \tag{2.20}$$

Besides the two-view geometry, there is three-view geometry in which three points $\mathbf{u}^{C1}$, $\mathbf{u}^{C2}$, and $\mathbf{u}^{C3}$, in three images respectively, representing the same point in the scene as illustrated in figure 2.5. There is a relationship between the three points and a $3 \times 3 \times 3$ tensor called trifocal tensor $\mathbf{T}$ as shown in equation

(2.21).

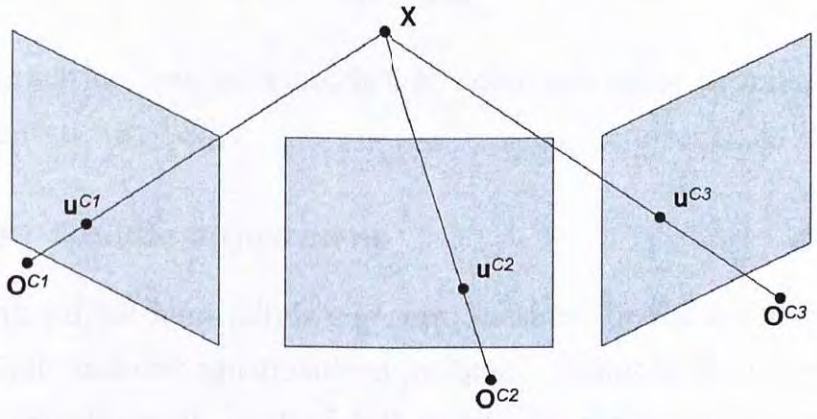

Figure 2.5: Illustration of the trifocal tensor.

$$[\mathbf{u}^{C2}]_\times (\sum_i u^{C1^i} T_i)[\mathbf{u}^{C3}]_\times = \mathbf{0}_{3\times 3} \tag{2.21}$$

To calculate the trifocal tensor $\mathbf{T}$, 7 corresponding points in three views are required.

It is generalized to N-view geometry for an arbitrary N. In N-view geometry, some computation methods like bundle adjustment are usually involved to refine the solution.

2.3.2 Factorization

Using factorization to solve the structure and motion problem was proposed by Tomasi and Kanade [31]. The projections of the 3-D points on the images are encapsulated in a matrix called measurement matrix $\mathbf{W}$. Based on the rank theory, factorization is applied to the measurement matrix $\mathbf{W}$ in equation (2.22) to get the motion matrix $\mathbf{M}$, representing the motion, and the

shape matrix $\mathbf{S}$, representing the 3-D structure.

$$\mathbf{W} = \mathbf{MS} \tag{2.22}$$

Factorization was later studied to solve the same problem in a sequential way [22].

2.3.3 Bundle adjustment

Bundle adjustment [33] is a general method for the solution of a multiple variable optimization problem. Suppose that we have the projections of a set of 3-D points on the images taken by several cameras and want to use the bundle adjustment to find out all the camera projection matrices and the 3-D coordinates of all the points in the scene, the re-projection errors between the observed and predicted positions of the points on the images (equation (2.23)) are minimized.

$$\sum_{i,j} d(\mathbf{P}^i \mathbf{X}_j, \mathbf{x}_j^i)^2 \tag{2.23}$$

In equation (2.23), $\mathbf{P}^i$ is the ith camera projection matrix, $\mathbf{X}_j$ is the jth 3-D point, and $\mathbf{x}_j^i$ is the projection of the jth 3-D point on the image taken by the ith camera. Newton's method and Levenberg-Marquardt method are usually used to minimize equation (2.23) in bundle adjustment.

Bundle adjustment is slow and the solution is accurate. It usually acts as a further refinement after other methods.

2.4 Sequential method

Sequential methods are able to recover the current camera pose right after the image has been taken. When a camera is moving and taking images, there would be connections between the neighbouring images in the image sequence. Sequential methods would use the information like the velocity of the camera in these connections to compute the result.

Brorida, Chandrashekhar, and Chellappa [6] used a full covariance iterated extended Kalman filter to recover the 3-D structure and the pose of the object from the image sequence taken by a single camera. Then Azarbayejani and Pentland [2] extended the work [6] by recovering the focal length of the camera as well as the structure and motion. They did so by adding one more parameter representing the focal length into the state of the extended Kalman filter.

Yu extended the work [2] by breaking the process into two stages [36]. One stage is pose updating handled by one extended Kalman filter, and the other is structure updating handled by N extended Kalman filters where N is the number of model points. The approach is more efficient than the work [2].

All the mentioned methods above compute the motion as well as the structure. There is a model-less pose tracking approach belonging to a different class. It is flexible and efficient since both the prior knowledge and computation of the structure are not required. Soatto et. al. [26] applied the essential constraint in epipolar geometry (two-view geometry) together with the Kalman filter to compute the pose information directly

from the image sequence. However, the essential constraint may become degenerate [13]. Yu used the trifocal tensor constraint [37] instead to make the system more robust. They further applied similar idea to recover the pose of a pair of stereo cameras [37].

The work in the thesis starts from the existing work by Yu [38]. The existing work [38] proposed an approach to recover the orientation and location of a pair of stereo cameras from the image sequences. An extended Kalman filter whose dynamic model is modeled by an approximated twist motion model was applied to track the pose of the stereo camera system. In the filter, trifocal tensor constraints are used to bypass the calculation of the structure to estimate the pose.

2.5 SLAM using cameras

Simultaneous localization and mapping (SLAM) does not belong to the field of computer vision, but the field of robotics. The target of SLAM is to locate a moving system consisting of a number of different sensors and develop a map of the surrounding scene simultaneously. However, some recent research [9] [23] [24] tried to use a camera as the only sensor since the data obtained from the camera is rich. One camera can be a substitute for many other sensors. However, processing is required for the data obtained from the camera. The problem of SLAM using cameras is similar to the structure and motion problem since both inputs are images and both outputs are structure and motion.

The approach proposed by Pupilli and Calway [23] is based

on both the particle filter [1] and unscented Kalman filter [17]. A particle filter is used for tracking the camera pose while an unscented Kalman filter coupled to the particle filter is used for estimating the structure.

The advantage of the approach is that it can survive even when there is a dynamic clutter in the image caused by sudden movement of the camera. The particle filter can detect such erratic motions and pause the structure estimation to prevent erroneous processing. Structure estimation would resume its processing after the motion has become normal.

The approach proposed by Davison [9] is based on the Kalman filtering. The orientation and location of the camera and the 3-D map of the scene are represented by a state vector $\mathbf{x}$ and a covariance matrix $\mathbf{P}$ which are used in both the extended Kalman filter and feature management.

The research work mainly focuses on feature measurement. Their approach manages features using an active approach. A matrix $\mathbf{S}$ is defined for each feature to represent the shape of a 2-D Gaussian probability density function over image coordinates. It can be used to define the possible area where the feature would lay on to enhance the efficiency and decrease the probability of mismatches. Furthermore, the matrix $\mathbf{S}$ can also be used to measure the contribution of the feature. Therefore, features with few contributions can be ignored to maintain a good set of features.

2.6 Summary

In this chapter, we have provided the background knowledge for this thesis and have reviewed the research work related to camera pose estimation.

□ **End of chapter.**

Chapter 3

Pose tracking of a stereo camera system

3.1 Overview

In this chapter, we shall investigate pose tracking of a pair of stereo cameras. Given the stereo image sequences from a pair of stereo cameras, we want to estimate the orientation and location of the stereo camera system.

3.1.1 Related work

There is some existing work related to the proposed approach in this chapter. The study in this chapter starts from the existing work [38] which proposed an approach to recover the pose of a stereo camera system from the stereo image sequences. They used an extended Kalman filter, whose dynamic model is modeled by an approximated twist motion model, with trifocal tensor constraints to track the pose of the stereo camera system without computing the structure explicitly.

The existing work [38] is based on the extended Kalman filter [11]. The filter handles a non-linear system by using the Jacobian to propagate the state and covariance of the system, so the state and covariance can only be accurate to the first order. The unscented Kalman filter [18] [16] proposed by Julier et. al. is another option to handle a non-linear dynamic system. The filter propagates the state and covariance using the unscented transform. The mean and covariance can be accurate to the second order. Therefore, the unscented Kalman filter can handle a non-linear system better than the extended Kalman filter. The unscented Kalman filter has been applied to several computer vision problems, which include vision based simultaneous localization and mapping (SLAM) [8] [23] [19], hand tracking [27], and eye tracking [39].

Besides the Kalman filters, evolutionary algorithms have also been applied in estimating the intrinsic and extrinsic parameters of a camera. Hati and Sengupta [14] computed the extrinsic parameters of a camera using a genetic algorithm. Ji and Zhang [15] also used a genetic algorithm to calibrate a camera. Cerveri et al. [7] used an enhanced evolutionary search to calibrate a stereo camera system. Toyama et al. [32] solved the problem of model-based pose estimation using a genetic algorithm. Yu et al. [35] improved Hati and Sengupta's work by incorporating a feature searching strategy to reject outliers.

3.1.2 Contribution

In this chapter, we propose an algorithm to track the pose of a stereo camera system.

- Our algorithm can recover more accurate pose than the existing work by using the Rodrigues' formula, in which no approximations are taken.
- Performances of different estimation methods including the extended Kalman filter [11], which handles a non-linear system by assuming local linearity, the unscented Kalman filter [34], which handles a non-linear system by statistical calculations, and the differential evolution [28], which is an instance of evolution algorithms having the property of good convergence, used in our approach are compared and analyzed. Advantages and disadvantages of them are discussed.

3.2 Problem definition

The geometry model is illustrated in figure 3.1. The projection of the mth 3-D model point $\mathbf{X}_m^O$ (homogeneous coordinate) on the image plane of camera i is $\mathbf{u}_{m,t}^{Ci}$ (homogeneous coordinate) at time t. The relationships are shown in equation (3.1).

$$\begin{aligned} \mathbf{u}_{m,t}^{C1} &= \mathbf{K}\begin{bmatrix}\mathbf{I}_{3\times3} & \mathbf{0}_{3\times1}\end{bmatrix}\mathbf{M}_t\mathbf{X}_m^O \\ \mathbf{u}_{m,t}^{C2} &= \mathbf{K}\begin{bmatrix}\mathbf{I}_{3\times3} & \mathbf{0}_{3\times1}\end{bmatrix}\mathbf{B}_{12}\mathbf{M}_t\mathbf{X}_m^O \end{aligned} \tag{3.1}$$

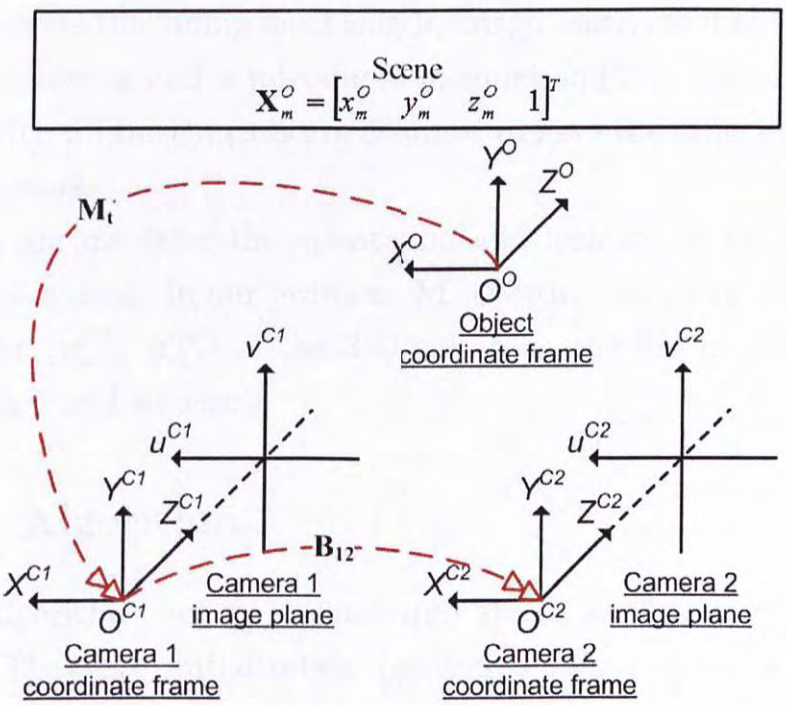

Figure 3.1: The image formation model of the stereo camera system.

The object coordinate frame is the reference coordinate frame. It is the same as the camera 1 coordinate frame at time 0. $\mathbf{M}_t$ is a 4×4 matrix that transforms 3-D model points $\mathbf{X}_m^O$ from the object (reference) coordinate frame to the camera 1 coordinate frame at time t. $\mathbf{B}_{12}$ is a 4×4 matrix that represents the rigid transformations from the coordinate frame of camera 1 to that of camera 2. $\mathbf{K}$ is a 3×3 matrix that encapsulates the intrinsic parameters (including focal length, image center, and pixel size) of the cameras and is introduced in equation (2.3). For the sake of clarity, all the cameras are assumed to have the same intrinsic parameters.

$\mathbf{M}_t$ encapsulates the orientation and location of the stereo camera system. In our problem, $\mathbf{M}_t$ is estimated given the projections ($\mathbf{u}_{m,t}^{C1}$, $\mathbf{u}_{m,t}^{C2}$) of the 3-D points on the image planes of camera 1 and camera 2.

3.3 Algorithm

The algorithm can be divided into stages as shown in figure 3.2. They are initialization (section 3.3.1), feature tracking and stereo correspondences matching (section 3.3.2), and pose tracking. Different estimation methods including the extended Kalman filter (section 3.3.4), unscented Kalman filter (section 3.3.5), and differential evolution (section 3.3.6) have been applied to pose tracking of the stereo camera system based on trifocal tensor constraints (section 3.3.3).

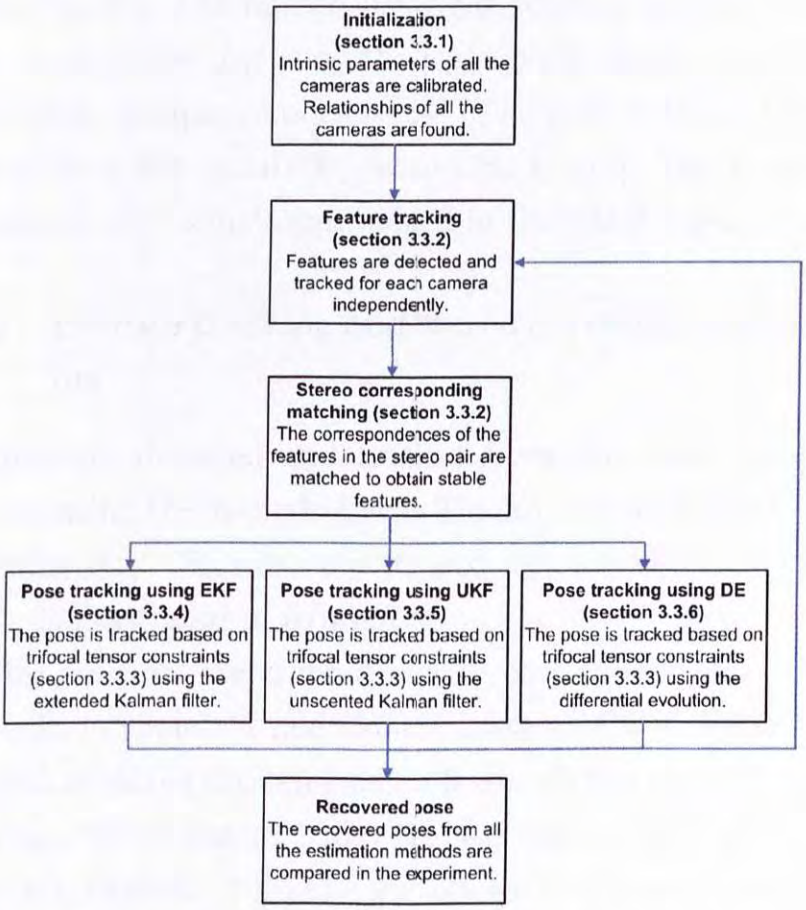

Figure 3.2: The overall algorithm for pose tracking of the stereo camera system.

3.3.1 Initialization

There are parameters that are required to be found in the initialization. The intrinsic parameters $\mathbf{K}$ of each camera are calibrated using the camera calibration toolbox [5]. The fundamental matrix $\mathbf{F}_{12}$ is computed using the toolbox [21] at the frame 0. It is calculated by using the eight-point algorithm [13] with the random sample consensus [10]. The matrix $\mathbf{B}_{12}$ is then calculated from the matrix $\mathbf{F}_{12}$ according to [13]. The matrix $\mathbf{F}_{12}$ and matrix $\mathbf{B}_{12}$ remain unchanged in the latter frames.

3.3.2 Feature tracking and stereo correspondence matching

Features are detected and tracked from the stereo image sequences using the Kanade-Lucas-Tomasi feature tracker [30] at each time step. Features are tracked for each camera independently as illustrated in figure 3.3.

After features are tracked, stereo correspondences between images from camera 1 and camera 2 are matched. Features are matched as stereo correspondence if the ith feature $(u_{i,t}^{C1}, v_{i,t}^{C1})$ in the image from camera 1 and the jth feature $(u_{j,t}^{C2}, v_{j,t}^{C2})$ in the image from camera 2 satisfy the following two conditions.

1. The distance between the epipolar line of the ith features in the image from camera 1 and the jth feature in the image from camera 2 is below a threshold D as shown in equation

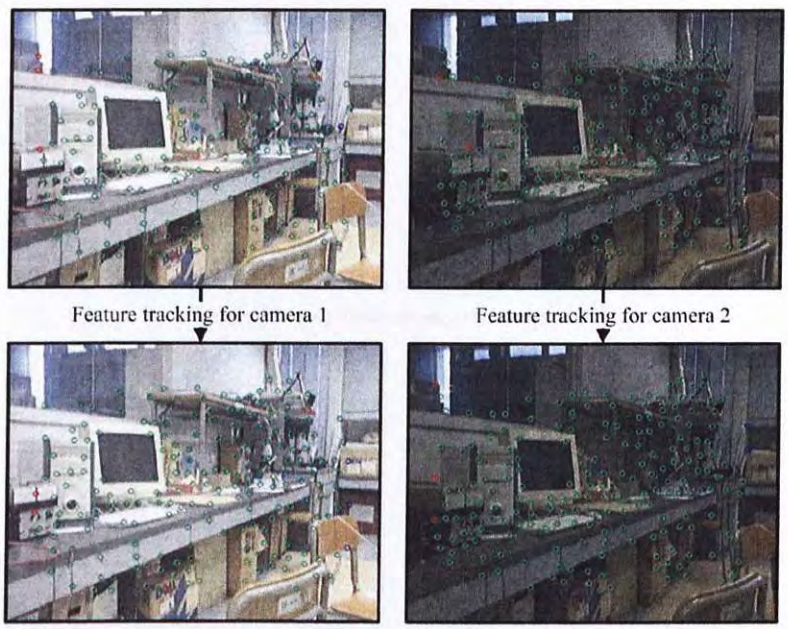

Figure 3.3: Illustration of feature tracking in pose tracking of the stereo camera system.

(3.2).

$$\begin{bmatrix} u_{j,t}^{C2} \\ v_{j,t}^{C2} \\ 1 \end{bmatrix} \mathbf{F}_{12} \begin{bmatrix} u_{i,t}^{C1} \\ v_{i,t}^{C1} \\ 1 \end{bmatrix} \leq D \tag{3.2}$$

2. The templates of the ith feature in the image from camera 1 and the jth feature in the image from camera 2 have large normalized cross-correlation value.

Features without correspondence are rejected as outliers to maintain a set of reliable features. Figure 3.4 shows an example of stereo correspondence matching.

Efficiency of the process can be maintained by using the following scheme. Firstly, the epipolar line of the ith feature in

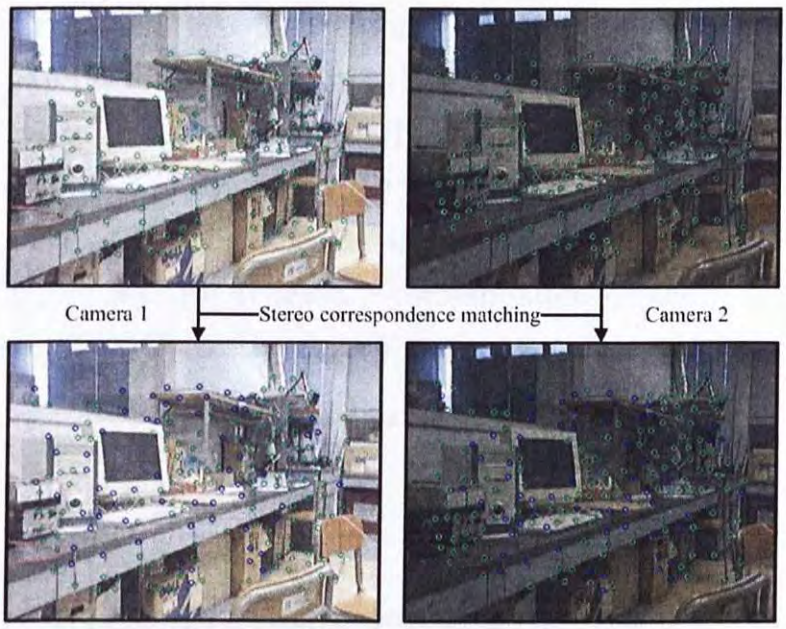

Figure 3.4: Illustration of stereo correspondence matching in pose tracking of the stereo camera system.

the image from camera 1 is calculated. Its correspondence in the image from camera 2 can then be searched by using normalized cross correlation within the threshold D from the epipolar line.

3.3.3 Pose tracking based on two trifocal tensors

In pose tracking, two trifocal tensors are used to relate the positions of the image points in four views illustrated in figure 3.5. One time step from the stereo image sequences is considered as a base time step. Initially, the first time step (i.e. time 0) is considered as the base time step. Both trifocal tensors use the stereo image pair at the base time step as their first two views. The image captured by camera 1 at time t is consid-

ered as the third view for tensor 1 $\mathbf{T}^1$. Matched points $\mathbf{u}^{C1}_{m,base}$, $\mathbf{u}^{C2}_{m,base}$ and $\mathbf{u}^{C1}_{m,t}$ in these three views are related by tensor 1 $\mathbf{T}^1$, where $m \in (1, \ldots, N)$. Similarly, the image captured by camera 2 at time t is considered as the third view for tensor 2 $\mathbf{T}^2$, which relates matched points $\mathbf{u}^{C1}_{m,base}$, $\mathbf{u}^{C2}_{m,base}$ and $\mathbf{u}^{C2}_{m,t}$, where $m \in (1, \ldots, N)$. As at least 7 correspondences are required to calculate the trifocal tensors, N must be at least 7.

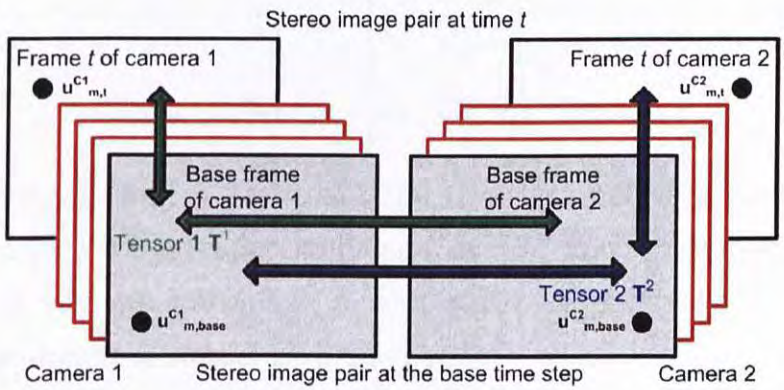

Figure 3.5: Illustration of the use of two trifocal tensors in pose tracking of the stereo camera system.

However, features may disappear and new features may appear when the stereo camera system moves. The number of correspondences N may be less than 7 in the four views. In this situation, the base time step needs to be reset. The time step $t-1$ become a new base time step. For the sake of clarity, we assume that there are no changes of the base time step required in section 3.3.4, section 3.3.5, and section 3.3.6.

3.3.4 Pose tracking using extended Kalman filter (Our EKF-2 approach)

The extended Kalman filter (EKF) is used to estimate the state of the stereo camera system at each time step.

The *state* vector $\mathbf{x}_t$ is defined as

$$\mathbf{x}_t = \begin{bmatrix} \dot{\mathbf{T}}_t^T & \dot{\mathbf{w}}_t^T \end{bmatrix}^T \tag{3.3}$$

$$\dot{\mathbf{T}}_t = \begin{bmatrix} \dot{x}_t & \dot{y}_t & \dot{z}_t \end{bmatrix}^T \tag{3.4}$$

$$\dot{\mathbf{w}}_t = \begin{bmatrix} \dot{\alpha}_t & \dot{\beta}_t & \dot{\gamma}_t \end{bmatrix}^T \tag{3.5}$$

where $\dot{x}_t$, $\dot{y}_t$, and $\dot{z}_t$ represent the translational velocities along x, y, and z-axes respectively and $\dot{\alpha}_t$, $\dot{\beta}_t$, and $\dot{\gamma}_t$ represent the angular velocities about x, y, and z-axes respectively.

The *dynamic model* is defined as

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \mathbf{v} \tag{3.6}$$

$\mathbf{v}$ is a 6×1 vector representing Gaussian process noise which models the changes of the velocities of the stereo camera system. The pose $\mathbf{M}_t$ of the stereo camera system is computed using

$$\begin{aligned} \mathbf{M}_t &= \begin{bmatrix} e^{\tilde{\dot{\mathbf{w}}}_t} & \dot{\mathbf{T}}_t \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \mathbf{M}_{t-1} \\ &= \begin{bmatrix} e^{\tilde{\dot{\mathbf{w}}}_t} & \dot{\mathbf{T}}_t \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{t-1} & \mathbf{T}_{t-1} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \end{aligned} \tag{3.7}$$

where

$$\tilde{\dot{\mathbf{w}}}_t = \begin{bmatrix} 0 & -\dot{\gamma}_t & \dot{\beta}_t \\ \dot{\gamma}_t & 0 & -\dot{\alpha}_t \\ -\dot{\beta}_t & \dot{\alpha}_t & 0 \end{bmatrix} \tag{3.8}$$

The exponential of the skew-symmetric matrix $\tilde{\dot{\mathbf{w}}}_t$ can be calculated using the Rodrigues' formula (3.9).

$$e^{\tilde{\dot{\mathbf{w}}}_t} = \mathbf{I} + \frac{e^{\tilde{\dot{\mathbf{w}}}_t}}{\|e^{\tilde{\dot{\mathbf{w}}}_t}\|} \sin \|e^{\tilde{\dot{\mathbf{w}}}_t}\| + \frac{e^{\tilde{\dot{\mathbf{w}}}_t}{}^2}{\|e^{\tilde{\dot{\mathbf{w}}}_t}\|^2}(1 - \cos \|e^{\tilde{\dot{\mathbf{w}}}_t}\|) \tag{3.9}$$

It is more precise than the approximated twist model (equation 3.10) used in [38] as no approximations are involved.

$$e^{\tilde{\dot{\mathbf{w}}}_t} = \mathbf{I} + \tilde{\dot{\mathbf{w}}}_t + \frac{\tilde{\dot{\mathbf{w}}}_t^{\,2}}{2!} + \frac{\tilde{\dot{\mathbf{w}}}_t^{\,3}}{3!} + \ldots \approx \mathbf{I} + \tilde{\dot{\mathbf{w}}}_t \tag{3.10}$$

The *measurement model* is defined as

$$\mathbf{y}_t = \mathbf{h}_t(\mathbf{x}_t) + \mathbf{n} = \mathbf{g}_t(\mathbf{M}_t) + \mathbf{n} \tag{3.11}$$

$\mathbf{n}$ is a $4N \times 1$ vector representing Gaussian measurement noise where N is the number of correspondences available for computing the pose. $\mathbf{g}_t(\mathbf{M}_t)$ is a function to compute the projections of all the available feature points on the image planes of camera 1

and camera 2 at time t as shown in equation (3.12).

$$\mathbf{g}_t(\mathbf{M}_t) = \begin{pmatrix} \begin{pmatrix} u_{1,t}^{C1} \\ v_{1,t}^{C1} \\ \vdots \\ u_{m,t}^{C1} \\ v_{m,t}^{C1} \\ \vdots \\ u_{N,t}^{C1} \\ v_{N,t}^{C1} \end{pmatrix}^T & \begin{pmatrix} u_{1,t}^{C2} \\ v_{1,t}^{C2} \\ \vdots \\ u_{m,t}^{C2} \\ v_{m,t}^{C2} \\ \vdots \\ u_{N,t}^{C2} \\ v_{N,t}^{C2} \end{pmatrix}^T \end{pmatrix}^T \tag{3.12}$$

where $(u_{m,t}^{C1}, v_{m,t}^{C1})$ and $(u_{m,t}^{C2}, v_{m,t}^{C2})$ are the stereo correspondences in the views from camera 1 and camera 2 at time t.

Based on the point transfer using the two trifocal tensors illustrated in figure 3.5, $\mathbf{g}_t(\mathbf{M}_t)$ is calculated using equation (3.13) and equation (3.14) represented in tensor notation. Details can be found in [13].

$$(U_{m,t}^{C1})^k = (U_{m,base}^{C1})^i (l_{m,base}^{C2})_j (T^1)_i^{jk} \tag{3.13}$$

$$(U_{m,t}^{C2})^k = (U_{m,base}^{C1})^i (l_{m,base}^{C2})_j (T^2)_i^{jk} \tag{3.14}$$

$\mathbf{U}_{m,t}^{Ci}$ is computed according to equation (3.15) to remove the effects of the intrinsic parameters of the cameras.

$$\mathbf{U}_{m,t}^{Ci} = \begin{pmatrix} (U_{m,t}^{Ci})^1 \\ (U_{m,t}^{Ci})^2 \\ (U_{m,t}^{Ci})^3 \end{pmatrix} = \mathbf{K}^{-1} \begin{pmatrix} u_{m,t}^{Ci} \\ v_{m,t}^{Ci} \\ 1 \end{pmatrix} \tag{3.15}$$

$\mathbf{l}^{C2}_{m,base}$ is a line passing through the mth feature point on the image plane of camera 2 at the base time step and can be found according to equation (3.16).

$$\mathbf{l}^{C2}_{m,base} = \begin{pmatrix} (l^{C2}_{m,base})_1 \\ (l^{C2}_{m,base})_2 \\ (l^{C2}_{m,base})_3 \end{pmatrix} = \begin{pmatrix} (l_{m,base})_2 \\ -(l_{m,base})_1 \\ -(U^{C2}_{m,t})^1(l_{m,base})_2 + (U^{C2}_{m,t})^2(l_{m,base})_1 \end{pmatrix}$$

$$\mathbf{l}_{m,base_e} = \begin{pmatrix} (l_{m,base})_1 \\ (l_{m,base})_2 \\ (l_{m,base})_3 \end{pmatrix} = \mathbf{e}_{12} \times \mathbf{U}^{C2}_{m,base} \tag{3.16}$$

where $\mathbf{e}_{12}$ is the epipole observed from camera 2 and $\mathbf{l}_{m,base_e}$ is the epipolar line passing through the mth feature point on the image plane of camera 2 at the base time step.

$(T^1)^{jk}_i$ and $(T^2)^{jk}_i$ in equation (3.13) and equation (3.14) represent the elements at the position (i, j, k) of $\mathbf{T}^1$ and $\mathbf{T}^2$ respectively. $\mathbf{T}^1$ and $\mathbf{T}^2$ are the two trifocal tensors illustrated in figure 3.5. Consider that $[\mathbf{I}_{3\times3}\ \mathbf{0}_{3\times1}]$ and $[\mathbf{I}_{3\times3}\ \mathbf{0}_{3\times1}]\mathbf{B}_{12} = [b^j_i]$ are the extrinsic parameters of camera 1 and camera 2 at the base time step respectively and $[\mathbf{I}_{3\times3}\ \mathbf{0}_{3\times1}]\mathbf{M}_t = [a^j_i]$ and $[\mathbf{I}_{3\times3}\ \mathbf{0}_{3\times1}]\mathbf{B}_{12}\mathbf{M}_t = [a'^{ij}_i])$ are the extrinsic parameters of camera 1 and camera 2 at time t respectively, $\mathbf{T}^1$ and $\mathbf{T}^2$ can be computed using equation (3.17).

$$\begin{aligned} T^{1jk}_i &= a^j_i b^k_4 - a^j_4 b^k_i \\ T^{2jk}_i &= a'^{ij}_i b^k_4 - a'^{ij}_4 b^k_i \end{aligned} \tag{3.17}$$

With the state, dynamic model, and measurement model defined, equations required for the EKF are derived according to [11] as follows.

Time update equations are

$$\begin{aligned} \hat{\mathbf{x}}_t^- &= \hat{\mathbf{x}}_{t-1} \\ \mathbf{P}_t^- &= \mathbf{P}_{t-1} + \mathbf{R}^{\mathbf{v}} \end{aligned} \tag{3.18}$$

Measurement update equations are

$$\begin{aligned} \mathbf{K}_t &= \mathbf{P}_t^- \nabla \mathbf{h}_t|_{\hat{\mathbf{x}}_t^-}{}^T (\nabla \mathbf{h}_t|_{\hat{\mathbf{x}}_t^-} \mathbf{P}_t^- \nabla \mathbf{h}_t|_{\hat{\mathbf{x}}_t^-}{}^T + \mathbf{R}^{\mathbf{n}})^{-1} \\ \hat{\mathbf{x}}_t &= \hat{\mathbf{x}}_t^- + \mathbf{K}_t(\mathbf{y}_t - \mathbf{h}_t(\hat{\mathbf{x}}_t^-)) \\ \mathbf{P}_t &= (\mathbf{I} - \mathbf{K}_t \nabla \mathbf{h}_t|_{\hat{\mathbf{x}}_t^-}) \mathbf{P}_t^- \end{aligned} \tag{3.19}$$

$\hat{\mathbf{x}}_t^-$ and $\hat{\mathbf{x}}_t$ are the states after time update and measurement update respectively. $\mathbf{P}_t^-$ and $\mathbf{P}_t$ are the 6 x 6 covariance matrices of the state $\hat{\mathbf{x}}_t^-$ and $\hat{\mathbf{x}}_t$ respectively. $\mathbf{R}^{\mathbf{v}}$ and $\mathbf{R}^{\mathbf{n}}$ are the 6×6 covariance matrices of the process noise $\mathbf{v}$ and the measurement noise $\mathbf{n}$ respectively. $\nabla \mathbf{h}_t|_{\hat{\mathbf{x}}_t^-}$ is the Jacobian matrix of the measurement equation $\mathbf{h}_t(\mathbf{x})$ at $\hat{\mathbf{x}}_t^-$. $\mathbf{K}_t$ is the $6 \times 4N$ Kalman gain matrix in the extended Kalman filter.

3.3.5 Pose tracking using unscented Kalman filter (Our UKF-2 approach)

Unscented Kalman filter (UKF) is used to estimate the state of the stereo camera system at each time step. The definitions of the state, dynamic model, and measurement model are the same as those used in the extended Kalman filter described in section 3.3.4. The equations required for the UKF are derived

according to [34] as follows.

Time update equations are

$$\begin{aligned}
\hat{\mathbf{x}}_t^- &= \hat{\mathbf{x}}_{t-1} \\
\mathbf{P}_t^- &= \mathbf{P}_{t-1} + \mathbf{R}^{\mathbf{v}} \\
\mathbf{X}_{t|t-1} &= \left[\hat{\mathbf{x}}_t^- \quad \hat{\mathbf{x}}_t^- + \sqrt{(L+\lambda)\mathbf{P}_t^-} \quad \hat{\mathbf{x}}_t^- - \sqrt{(L+\lambda)\mathbf{P}_t^-}\right] \\
\mathbf{Y}_{t|t-1} &= \mathbf{h}_t^*(\mathbf{X}_{t|t-1}) \\
\hat{\mathbf{y}}_t^- &= \sum_{i=0}^{2L} W_i^{(m)} Y_{i,t|t-1}
\end{aligned} \tag{3.20}$$

Measurement update equations are

$$\begin{aligned}
\mathbf{P}_{\tilde{\mathbf{y}}_t\tilde{\mathbf{y}}_t} &= \sum_{i=0}^{2L} W_i^{(c)} (Y_{i,t|t-1} - \hat{\mathbf{y}}_t^-)(Y_{i,t|t-1} - \hat{\mathbf{y}}_t^-)^T + \mathbf{R}^{\mathbf{n}} \\
\mathbf{P}_{\mathbf{x}_t\mathbf{y}_t} &= \sum_{i=0}^{2L} W_i^{(c)} (X_{i,t|t-1} - \hat{\mathbf{x}}_t^-)(Y_{i,t|t-1} - \hat{\mathbf{y}}_t^-)^T \\
\mathbf{K}_t &= \mathbf{P}_{\mathbf{x}_t\mathbf{y}_t} \mathbf{P}_{\tilde{\mathbf{y}}_t\tilde{\mathbf{y}}_t}{}^{-1} \\
\hat{\mathbf{x}}_t &= \hat{\mathbf{x}}_t^- + \mathbf{K}_t(\mathbf{y}_t - \hat{\mathbf{y}}_t^-) \\
\mathbf{P}_t &= \mathbf{P}_t^- - \mathbf{K}_t \mathbf{P}_{\tilde{\mathbf{y}}_t\tilde{\mathbf{y}}_t} \mathbf{K}_t{}^T
\end{aligned} \tag{3.21}$$

where

$$\begin{aligned}
W_0^{(m)} &= \frac{\lambda}{L+\lambda} \\
W_0^{(c)} &= \frac{\lambda}{L+\lambda} + 1 - \alpha^2 + \beta \\
W_i^{(m)} &= W_i^{(c)} = \frac{1}{2(L+\lambda)}, i = 1, 2, \ldots, 2L
\end{aligned} \tag{3.22}$$

λ is a scaling parameter. α and β represent the spread of the

sigma points and the prior knowledge of the distribution of the state $\mathbf{x}_t$ respectively. $\mathbf{X}_{t|t-1}$ contains all the sigma points used in the unscented transform while $\mathbf{X}_{i,t|t-1}$ indicates the ith sigma point. $L = 6$ is the dimension of the state $\mathbf{x}_t$. $W_0^{(m)}$ and $W_0^{(c)}$ are the weights used in calculating the mean and the covariance matrices respectively. $\mathbf{h}_t^*(\mathbf{X}_{t|t-1})$ is a function which calculates all $2L+1$ sigma points in $\mathbf{X}_{t|t-1}$ using $\mathbf{h}_t(\mathbf{x}_t)$ defined in equation (3.11). That is

$$\begin{aligned} &\mathbf{h}_t^*(\mathbf{X}_{t|t-1}) = \mathbf{Y}_{t|t-1} \\ &\text{means} \\ &\mathbf{h}_t(\mathbf{X}_{i,t|t-1}) = \mathbf{Y}_{i,t|t-1} \\ &i \in (1, 2, \ldots, 2L, 2L+1) \end{aligned} \tag{3.23}$$

$\hat{\mathbf{y}}_t^-$ is the predicted measurement computed from the unscented transform at time t. $\mathbf{P}_{\tilde{\mathbf{y}}_t\tilde{\mathbf{y}}_t}$ is the $4N \times 4N$ covariance matrix between the elements of the predicted measurement $\hat{\mathbf{y}}_t^-$. $\mathbf{P}_{\mathbf{x}_t\mathbf{y}_t}$ is the $6 \times 4N$ covariance matrix between the elements of the state $\hat{\mathbf{x}}_t^-$ and the predicted measurement $\hat{\mathbf{y}}_t^-$. Similar to the EKF, $\mathbf{R}^{\mathbf{v}}$ and $\mathbf{R}^{\mathbf{n}}$ are the 6×6 covariance matrices of the process noise $\mathbf{v}$ and the measurement noise $\mathbf{n}$ respectively. $\hat{\mathbf{x}}_t^-$ and $\hat{\mathbf{x}}_t$ are the states at time t after time update and measurement update respectively. $\mathbf{P}_t^-$ and $\mathbf{P}_t$ are the 6×6 covariance matrices of the states $\hat{\mathbf{x}}_t^-$ and $\hat{\mathbf{x}}_t$ respectively. $\mathbf{K}_t$ is the $6 \times 4N$ Kalman gain matrix at time t. The mean and covariance are propagated using the unscented transform in the UKF while they are propagated using the Jacobian in the EKF.

3.3.6 Pose tracking using differential evolution (Our DE-2 approach)

Differential evolution (DE) [29] [28], a instance of evolutionary algorithm having the property of good convergence, is used to estimate the pose of the stereo camera system at each time step. Figure 3.6 shows the outline of the differential evolution.

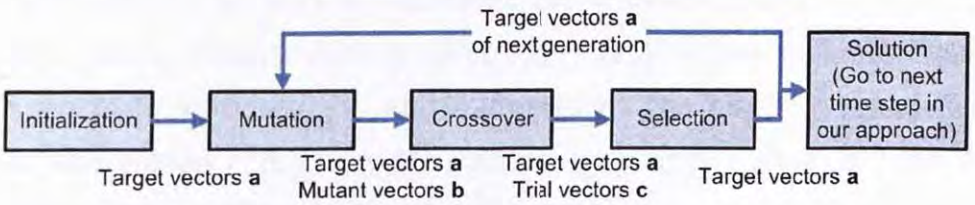

Figure 3.6: Outline of the differential evolution.

Initially, there is a set of target vectors (candidate solutions) **a**. In mutation, mutant vectors **b** are generated using the target vectors **a**. In crossover, trial vectors **c** are generated using the target vectors **a** and the mutant vectors **b**. Target vectors **a** of the next generation would be selected between the target vectors **a** of the current generation and the trial vectors **c** using a cost function. The iteration continues until

1. The cost function is minimized to a desired value, or
2. The number of generations reaches a desired value.

There are several variants of the DE. In the proposed approach, we use the DE scheme which is classified as *DE/rand/1/bin* strategy using the notation described in [29] [28].

In the proposed algorithm, each of target vectors **a**, mutant vectors **b**, and trial vectors **c** consists of a vector $\mathbf{x}_t$ and a vector

$\mathbf{k}_t$ as shown in figure 3.7. $\mathbf{x}_t$ consists of $\dot{x}_t$, $\dot{y}_t$, $\dot{z}_t$, $\dot{\alpha}_t$, $\dot{\beta}_t$, and

a								
$\mathbf{x}_t$						$\mathbf{k}_t$		
$\dot{x}_t$	$\dot{y}_t$	$\dot{z}_t$	$\dot{\alpha}_t$	$\dot{\beta}_t$	$\dot{\gamma}_t$	k_{t1}	...	k_{tn}

Figure 3.7: Format of target vectors, mutant vectors, and trial vectors used in the differential evolution for pose tracking of the stereo camera system.

$\dot{\gamma}_t$ where $\dot{x}_t$, $\dot{y}_t$, and $\dot{z}_t$ represent the translational velocities of the stereo camera system along x, y and z-axes at time t respectively while $\dot{\alpha}_t$, $\dot{\beta}_t$, and $\dot{\gamma}_t$ represent the angular velocities of the stereo camera system about x, y and z-axes at time t respectively. They are real numbers that are estimated using the differential evolution. As only the velocities are estimated, the search space is small and thus the efficiency of the approach is maintained. For the sake of efficiency, the pose $\mathbf{M}_t$ is calculated approximately from vector $\mathbf{x}_t$ by truncating the higher order terms of its Taylor expansion as shown in equation (3.24).

$$\begin{aligned}
\mathbf{M}_t &= \begin{bmatrix} e^{\tilde{\dot{\mathbf{w}}}_t} & \dot{\mathbf{T}}_t \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \mathbf{M}_{t-1} \\
&= \begin{bmatrix} \mathbf{I} + \tilde{\dot{\mathbf{w}}}_t + \frac{\tilde{\dot{\mathbf{w}}}_t^{\,2}}{2!} + \frac{\tilde{\dot{\mathbf{w}}}_t^{\,3}}{3!} + \ldots & \dot{\mathbf{T}}_t \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{t-1} & \mathbf{T}_{t-1} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \\
&\approx \begin{bmatrix} \mathbf{I} + \tilde{\dot{\mathbf{w}}}_t & \dot{\mathbf{T}}_t \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{t-1} & \mathbf{T}_{t-1} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}
\end{aligned} \tag{3.24}$$

where

$$\tilde{\dot{\mathbf{w}}}_t = \begin{bmatrix} 0 & -\dot{\gamma}_t & \dot{\beta}_t \\ \dot{\gamma}_t & 0 & -\dot{\alpha}_t \\ -\dot{\beta}_t & \dot{\alpha}_t & 0 \end{bmatrix}, \dot{\mathbf{T}}_t = \begin{bmatrix} \dot{x}_t \\ \dot{y}_t \\ \dot{z}_t \end{bmatrix} \tag{3.25}$$

$\mathbf{k}_t$ stores the distinct indexes of the correspondences used in pose tracking in ascending order. They are not estimated using the differential evolution directly due to their nature. In our approach, n is set to the smaller value of 20 and the number of correspondences to reject outliers and make the approach more stable and efficient.

In the initialization, NP target vectors are generated randomly.

In the operation of mutation, mutant vector $\mathbf{b}_{i,G+1}$ is generated from target vectors for each $i \in (1, 2, \ldots, NP)$.

Vector $\mathbf{x}_t$ inside mutant vector $\mathbf{b}_{i,G+1}$ is generated by using equation (3.26) to perform mutation.

$$\mathbf{b}^{\mathbf{x}}{}_{i,G+1} = \mathbf{a}^{\mathbf{x}}{}_{r_1,G} + F(\mathbf{a}^{\mathbf{x}}{}_{r_2,G} - \mathbf{a}^{\mathbf{x}}{}_{r_3,G}) \tag{3.26}$$

$r_1 \in \{1, 2.., NP\} \setminus \{r_2, r_3, i\}$, $r_2 \in \{1, 2.., NP\} \setminus (r_1, r_3, i)$, and $r_3 \in \{1, 2.., NP\} \setminus \{r_1, r_2, i\}$ are random generated indexes. $F \in [0, 2]$ is used to control the amplification of the difference of the target vectors.

Vector $\mathbf{k}_t$ inside mutant vector $\mathbf{b}_{i,G+1}$ is generated according to equation (3.27). Actually, no mutation is performed due to its nature.

$$\mathbf{b}^{\mathbf{k}}{}_{i,G+1} = \mathbf{a}^{\mathbf{k}}{}_{i,G} \tag{3.27}$$

In the operation of crossover, trial vector $\mathbf{c}_{i,G+1}$ is generated from target vectors and mutant vectors for each $i \in (1, 2, \ldots, NP)$.

Vector $\mathbf{x}_t$ inside trial vector $\mathbf{c}_{i,G+1}$ is generated by using equation (3.28) to randomly select elements from target vectors and mutant vectors.

$$\begin{aligned} &\mathbf{c^x}_{i,G+1} = (c^x_{1i,G+1}, c^x_{2i,G+1}, \ldots, c^x_{Di,G+1}) \\ &\text{where} \\ &c^x_{ji,G+1} = \begin{cases} b^x_{ji,G+1} & \text{if } r(j) \leq CR \text{ or } j = rn \\ a^x_{ji,G} & \text{if } r(j) > CR \text{ and } j \neq rn \end{cases} \end{aligned} \tag{3.28}$$

$r(j) \in [0, 1]$ is the jth evaluation of random real number. CR is the crossover constant controlling the rate of crossover. $rn \in \{1, 2, \ldots, D\}$ is a random generated index that enable $\mathbf{c}^x_{i,G+1}$ to get at least one element from $\mathbf{b}^x_{i,G+1}$.

Vector $\mathbf{k}_t$ inside trial vector $\mathbf{c}_{i,G+1}$ is generated by using equation (3.29) to randomly combine target vectors and mutant vectors.

$$\mathbf{c^k}_{i,G+1} = \begin{cases} (a^k_{1i}, a^k_{2i}, \ldots, a^k_{hi}, b^k_{(h+1)j}, \ldots, b^k_{nj}) & \text{if } a^k_{hi} \leq b^k_{hj} \\ (b^k_{1j}, b^k_{2j}, \ldots, b^k_{hj}, a^k_{(h+1)i}, \ldots, a^k_{ni}) & \text{otherwise} \end{cases}$$

$$\begin{aligned} &\text{where} \\ &\mathbf{a^k}_{i,G} = (a^k_{1i}, a^k_{2i}, \ldots, a^k_{ni}) \\ &\mathbf{b^k}_{j,G} = (b^k_{1j}, b^k_{2j}, \ldots, b^k_{nj}) \end{aligned} \tag{3.29}$$

$h \in \{1, 2, ..., n\}$ defining the crossover point of the two vectors and $j \in \{1, 2, ..., NP\}$ indicating the crossover vector are generated randomly.

In the operation of selection, NP target vectors are selected using the following scheme.

$$\mathbf{a}_{i,G+1} = \begin{cases} \mathbf{c}_{i,G+1} & \text{if } f(\mathbf{c}_{i,G+1}) < f(\mathbf{a}_{i,G}) \\ \mathbf{a}_{i,G} & \text{otherwise} \end{cases} \tag{3.30}$$

If the cost of the trial vector $\mathbf{c}_{i,G+1}$ is smaller, $\mathbf{c}_{i,G+1}$ is selected. Otherwise, $\mathbf{a}_{i,G}$ is kept to the next generation.

In our approach, the cost function in the differential evolution is defined as

$$\begin{aligned} f(\mathbf{a}) &= \|\mathbf{y}_t(\mathbf{k}_t) - \mathbf{h}_t(\mathbf{a})\| \\ &= \|\mathbf{y}_t(\mathbf{k}_t) - \mathbf{g}_t(\mathbf{M}_t, \mathbf{k}_t)\| \end{aligned} \tag{3.31}$$

which is the root-mean-square of the differences between the measured and estimated positions of the image points selected by the vector $\mathbf{k}_t$. $\mathbf{y}_t(\mathbf{k}_t)$ is the measured positions of the selected feature points in the images from camera 1 and camera 2 at time t while $\mathbf{g}_t(\mathbf{M}_t, \mathbf{k}_t)$ is a function to compute the estimated positions of the selected feature points on the image planes of camera 1 and camera 2 at time t. It is the same as equation (3.12) described in section 3.3.4 except only correspondences selected by $\mathbf{k}_t$ are involved.

In the synthetic and real experiments, the setting of parameters summarized in table 3.1 is used.

Table 3.1: Values of parameters used in the differential evolution.

Parameters	Values
Population size NP	60
Amplification of the difference between two vectors F	0.8
Constant to control crossover rate CF	0.8
Number of generations	300
Number of selected features	Smaller value of 20 and number of available correspondences

3.4 Experiment

3.4.1 Synthetic experiments

Four approaches were tested in the experiment using synthetic data. They included the EKF-2 approach, UKF-2 approach, and DE-2 approach of our algorithm for pose tracking of the stereo camera system. These three approaches were compared with the approach proposed by Yu et. al. [38] in which they made use of the extended Kalman filter with the trifocal tensor constraints and the approximated twist motion model. The details of the four approaches are summarized in table 3.2. All the approaches were implemented in Matlab 7.4 and run on a computer with Pentium D 3.0 GHz CPU and 1 GB RAM. The objective of the experiment is to compare the performances of these four methods in terms of accuracy and efficiency.

In the synthetic data experiment, two cameras with resolution 640 $\times$ 480 were used. Both of them had 4.6 mm focal

Table 3.2: List of approaches tested in the synthetic experiment of pose tracking of the stereo camera system.

Name	Desciption
Our EKF-2 approach	Our EKF-based approach for pose tracking of a pair of stereo cameras described in the section 3.3.4.
Our UKF-2 approach	Our UKF-based approach for pose tracking of a pair of stereo cameras described in the section 3.3.5.
Our DE-2 approach	Our differential evolution-based approach for pose tracking of a pair of stereo cameras described in the section 3.3.6.
Yu's approach [38]	The approach proposed by Yu et. al. [38]. It makes use of the extended Kalman filter with the trifocal tensor constraints and the approximated twist motion model.

length and a 2-D zero-mean Gaussian noise of 1 pixel standard deviation. Camera 1 and camera 2 were put 0.1 m apart and facing the same direction as illustrated in figure 3.8. Five hundred feature points were generated randomly in 3-D space at places $1 - 6$ m away from camera 1. The length of each test sequence was 90 frames. The motion of the stereo camera system consisted of three different segments. They included mixed motion (both rotation and translation) section, pure rotation section, and pure translation section. Each of them consisted of 30 frames. The motion was generated randomly with maximum translation ± 0.01 m along x, y, and z-axes and maximum rotation $\pm 1°$ about x, y, and z-axes per frame. Zero-mean Gaussian noises of translation 0.001 m and rotation 0.1° standard deviation were added to the motion parameters to simulate the non-

smoothness in the real world. 50 independent tests were carried out in the experiment.

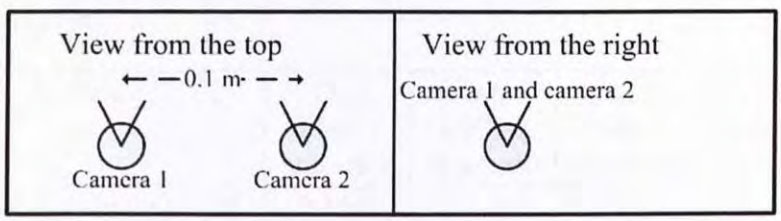

Figure 3.8: The setting of the stereo camera system in the synthetic experiment.

To compare the results, we extracted roll angle (rotation about x-axis), pitch angle (rotation about y-axis), and yaw angle (rotation about z-axis) from the recovered rotation $\hat{\mathbf{R}}$ to compare with those from the true rotation $\mathbf{R}$. The rotational errors in the tth frame of the ith test are computed using

$$Roll_{t,i}^{err} = |Roll_{\hat{\mathbf{R}}_{t,i}} - Roll_{\mathbf{R}_{t,i}}| \tag{3.32}$$

$$Pitch_{t,i}^{err} = |Pitch_{\hat{\mathbf{R}}_{t,i}} - Pitch_{\mathbf{R}_{t,i}}| \tag{3.33}$$

$$Yaw_{t,i}^{err} = |Yaw_{\hat{\mathbf{R}}_{t,i}} - Yaw_{\mathbf{R}_{t,i}}| \tag{3.34}$$

For the translation, we use the Euclidean norm of the difference between the true translation $\mathbf{T}$ and the recovered one $\hat{\mathbf{T}}$ for comparison. The translational error in the tth frame of the ith test is computed using

$$T_{t,i}^{err} = ||\mathbf{T}_{t,i} - \hat{\mathbf{T}}_{t,i}|| \tag{3.35}$$

Table 3.3 shows the average pose errors per frame, average number of correspondences per frame, and average processing time per frame of 50 independent tests.

Table 3.3: Results of the synthetic experiment of pose tracking of the stereo camera system.

Name	Our EKF-2 approach	Our UKF-2 approach	Our DE-2 approach	Yu's approach [38]
Average errors of roll angle per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} Roll_{t,i}^{err}}{T\times N}$ (degree)	0.125°	0.125°	0.249°	0.149°
Average errors of pitch angle per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} Pitch_{t,i}^{err}}{T\times N}$ (degree)	0.095°	0.095°	0.427°	0.114°
Average errors of yaw angle per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} Yaw_{t,i}^{err}}{T\times N}$ (degree)	0.293°	0.293°	0.548°	0.328°
Average errors of translation per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} T_{t,i}^{err}}{T\times N}$ (meter)	0.0158m	0.0158m	0.0891m	0.0196m
Average number of available correspondences per frame	24.55	24.55	24.55	24.55
Average processing time per frame (second)	0.04s	0.12s	57.30s	0.02s
N is the number of tests which is 50. T is the number of frames which is 90.				
All the algorithms were implemented in Matlab 7.4 and run on a computer with Pentium D 3.0 GHz CPU and 1 GB RAM.				

The first four rows of table 3.3 compare these approaches in terms of the three rotation angle errors and the root-mean-square translation error. They show that our EKF-2 and UKF-2 approach could recover more accurate pose than Yu's approach [38] since our EKF-2 and UKF-2 approaches use the Rodrigues' formula to represent rotation. It is more precise than the approximated twist motion model used in the Yu's approach. The processing time of our EKF-2 approach was slightly longer than Yu's approach as shown in the last row because its measurement model is more complex due to the Rodrigues' formula.

In the comparison between the EKF-2 approach and UKF-2 approach, both of them achieved similar accuracy which means that the assumption of the local linearity of the system is appropriate in the extended Kalman filter. The processing time of the UKF-2 approach was longer than that of the EKF-2 approach because the UKF-2 approach needs to compute the function $\mathbf{H}_t^*(\mathbf{X}_{t|t-1})$, which is a time consuming task, in the unscented transformation. As a result, the UKF-2 approach spends more time than the EKF-2 approach in propagating the mean and covariance while only direct Jacobian matrix calculation is required in the EKF-2 approach. However, there is an important issue about the comparison of time. In our implementation, the UKF-2 approach contains many looping statements which Matlab is weak for. The result may be different if the approach is implemented in other programming languages. Anyway, we can see that the UKF-2 approach can still work in real-time when there are less than 50 correspondences which are enough for recovering the pose information accurately.

In the synthetic experiment, it seems that the DE-2 approach does not have any advantages. Its processing time was long and its accuracy was low. However, the performances of differential evolution highly depend on the setting of the parameters used in the differential evolution. They include the number of generation G, population size NP, amplification of difference between two vectors F, crossover constant CR. The result should be different when different values of these parameters are set. It is trivial that more accurate result can be obtained if larger values of N and NP are used. However, it is also expected that the corresponding processing time must be longer. Anyway, when the DE-2 approach is compared with the EKF-2, UKF-2, and Yu's approach, we can observe that the differential evolution is not suitable for our problem. However, as the cost function in our DE-2 approach can be computed independently for each target vector and trial vector, it may have advantages when parallel computing is available.

As the DE-2 approach and Yu's approach make use of the approximated twist model, it is interesting to investigate its effect on the recovered rotation matrix $\mathbf{R}$. Figure 3.9 shows the determinant of the rotation matrix $\mathbf{R}$ against frame number. Normally, the determinant of the rotation matrix $\mathbf{R}$ must be one. From the figure, the determinants of the matrices $\mathbf{R}$ recovered by the DE-2 approach and Yu's approach deviate from one gradually. The reason is that the approximation assumes that the motion is very small. It would induce error in each time step especially when there are changes of orientation (first 60 frames). As a result, this kind of approximation should not be

taken.

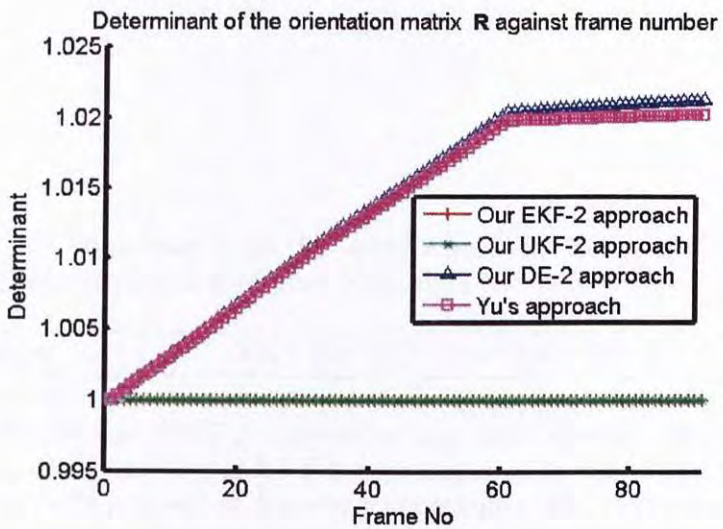

Figure 3.9: Determinant of the rotation matrix **R** against frame number in the synthetic experiment of pose tracking of the stereo camera system.

Table 3.4 and table 3.5 summarize the performances of all the tested algorithms in the synthetic experiment.

3.4.2 Real experiments

We have performed experiment using real stereo image sequences. Stereo image sequences with ground truth data were used to evaluate the performances of our EKF-2 approach, UKF-2 approach, and DE-2 approach in pose tracking of the stereo camera system. The objective of the experiment is to show that the algorithm can work accurately in the real world.

In the real experiment, two web cameras with resolution 320×240 shown in figure 3.11 were mounted on top of a robot shown in figure 3.10. The robot was driven by two servo motors that were attached to the wheels on the left and right.

Table 3.4: Comparison of all the tested algorithms in terms of accuracies in the synthetic experiment of pose tracking of the stereo camera system.

Accuracy:	EKF-2 $\approx$ UKF-2 > Yu's > DE-2
Explanation: Our EKF-2 and UKF-2 approaches are more accurate than Yu's approach. Our EKF-2 and UKF-2 approaches make use of the Rodrigues' formula, which does not have any approximations, to represent the rotation. So the approaches are better than Yu's approach, which uses approximations to represent the rotation. Our EKF-2 and UKF-2 approaches have similar accuracies. The UKF uses the unscented transform to propagate the mean and covariance of the system while the EKF propagates the mean and covariance by assuming local linearization. The UKF can achieve the second order accuracy in the Taylor series while the EKF can only achieve the first order accuracy. However, our experiment shows that both of them have similar accuracies for our problem. We believe that it is because the higher orders are not significant in our system. Therefore, the EKF is already enough for our system. The accuracy of our DE-2 approach is not very good. The major reason is that the parameter setting, which affects the accuracy, used in the experiment may not be a good one. Further investigation is required. However, the approach is much less efficient than our EKF-2 and UKF-2 approaches. Therefore, it may not be valuable to investigate.	

Table 3.5: Comparison of all the tested algorithms in terms of efficiencies in the synthetic experiment of pose tracking of the stereo camera system.

Efficiency:	Yu's > EKF-2 > UKF-2 > DE-2
Explanation: Our EKF-2 approach is slightly slower than Yu's approach because the Rodrigues' formula, used in our EKF-2 approach, is slightly more complex than the approximated twist motion model, used in Yu's approach. Our UKF-2 approach is slower than our EKF-2 approach because more time is spent on propagating the mean and covariance of the system in the UKF. Only the Jacobian calculation is required in the EKF while the unscented transform is needed in the UKF. The unscented transform needs more time because of computing the function $\mathbf{h}_t^*(\mathbf{X}_{t\|t-1})$. Our DE-2 approach is very slow because there are many computations of the cost function, which takes time, in each time step.	

A personal computer sent control signals to control its movements. To change the direction, two wheels were made to move at different motions. For instance, moving left motor forward and right motor backward could make the robot turn right at a certain degree. Images taken by the stereo camera system were transferred to the personal computer via Universal Serial Bus (USB). Given the diameters of the wheels, distance between them and robot displacement per motor step, we could compute the actual orientation and location of the stereo camera system (ground truth).

To compare the results, we extracted roll, pitch and yaw angles from the recovered rotation to make a comparison with those true angles. For the translation, we extracted translations along x-axis, y-axis and z-axis from the recovered translation to make a comparison with those true translations.

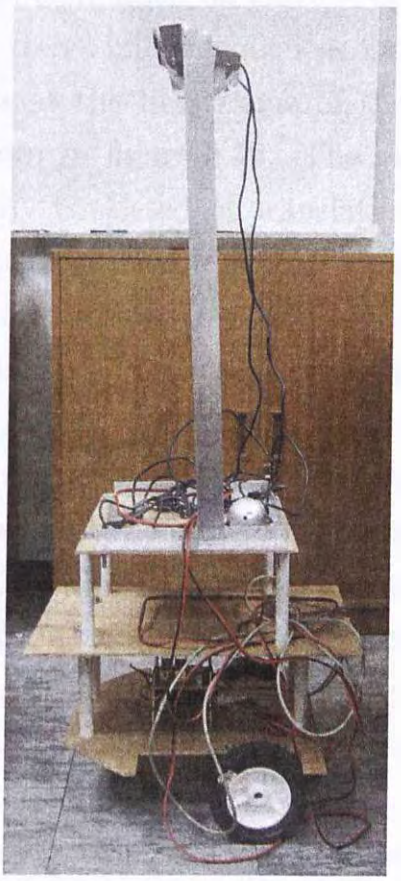

Figure 3.10: The robot on which the cameras are mounted in the real experiment.

Figure 3.11: The pair of stereo cameras mounted on the robot in the real experiment.

Experiment was conducted on three sets of stereo image sequences. The first stereo image sequences consisted of 120 frames. The stereo images at the first frame of the first stereo image sequences are shown in figure 3.12. The motion of the stereo camera system only consisted of translation. The number of the correspondences available for computing the motion of the stereo camera system was between 30 and 68. The recovered pose is shown in figure 3.15. Table 3.6 shows the timings of the experiment.

The second sequences consisted of 55 frames. The stereo images at the first frame of the second stereo image sequences are shown in figure 3.13. The motion of the stereo camera system only consisted of rotation. The number of the available correspondences was between 11 and 44. The experimental result is shown in figure 3.16. Table 3.7 shows the timings of the experiment.

The third sequences consisted of 155 frames. The stereo images at the first frame of the third sequences are shown in figure 3.14. The motion of the stereo camera system consisted of both translation and rotation. The number of the available features was between 7 and 53. Figure 3.17 shows the corresponding experimental result. Table 3.8 shows the timings of the experiment.

In figure 3.15, figure 3.16, and figure 3.17, lines with ($\bigcirc$) show the ground truth. Lines with ($+$) show the recovered pose by our EKF-2 approach while lines with ($\times$) show the recovered pose by our UKF-2 approach. Lines with ($\triangle$) show the recovered pose by our DE-2 approach. Comparing the ground truth data

and the recovered pose information by all of our approaches, we see that the recovered poses by our EKF-2 approach and UKF-2 approach of the proposed algorithms were accurate. The proposed algorithms work well in real situations. However, similar to the result obtained in the synthetic experiment, the pose information recovered by the DE-2 approach is not very accurate. It is mainly due to its value settings of the parameters in the differential evolution.

In table 4.11, table 4.12, and table 4.13, the first row shows the times used per frame in feature detection and tracking of each camera. The second row shows the time used per frame in stereo correspondence matching of the stereo pair. The third row shows the times used per frame in pose tracking of each tested algorithm.

Table 3.9 summarizes the results of the synthetic and real experiments.

Figure 3.12: The stereo images at the first frame of the first stereo image sequences in the real experiment of pose tracking of the stereo camera system. (Left) Image from camera 1. (Right) Image from camera 2.

Figure 3.13: The stereo images at the first frame of the second stereo image sequences in the real experiment of pose tracking of the stereo camera system. (Left) Image from camera 1. (Right) Image from camera 2.

Figure 3.14: The stereo images at the first frame of the third stereo image sequences in the real experiment of pose tracking of the stereo camera system. (Left) Image from camera 1. (Right) Image from camera 2.

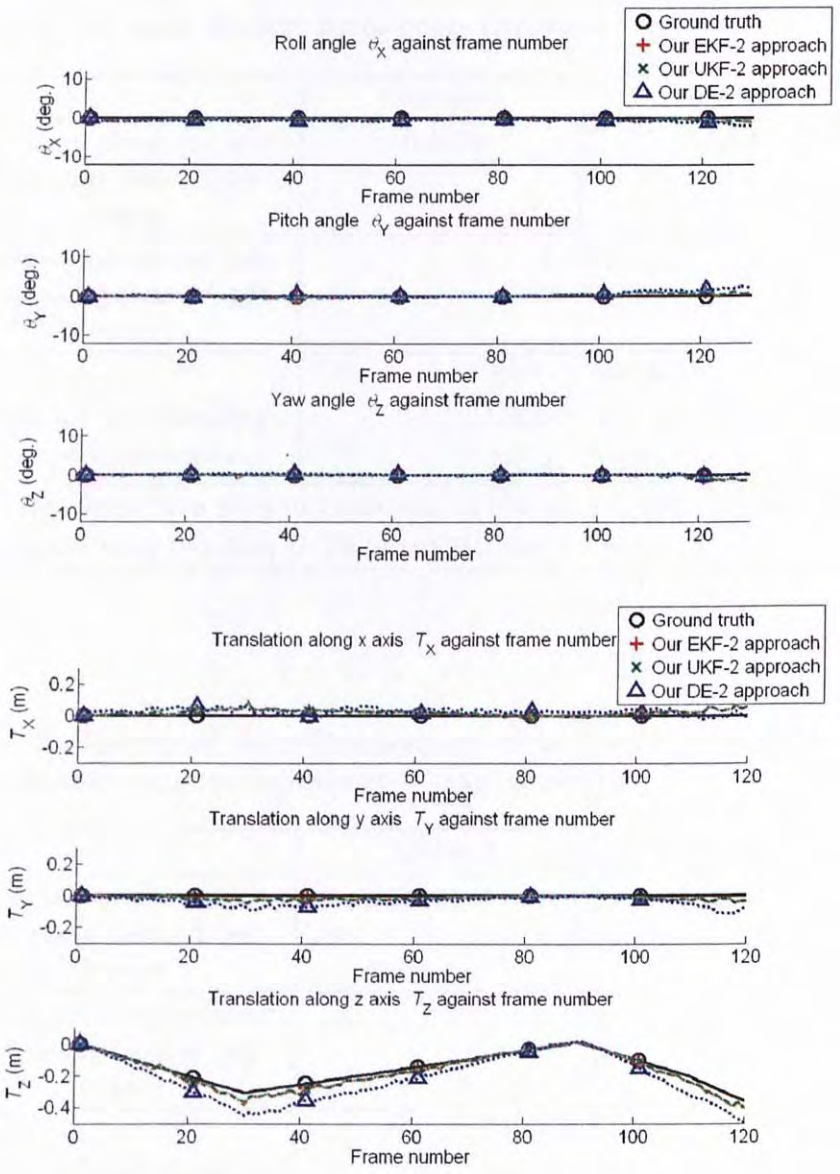

Figure 3.15: Result of the real experiment of pose tracking of the stereo camera system using the first stereo image sequences. (Top) Rotation. (Bottom) Translation.

Table 3.6: Timings of the real experiment of pose tracking of the stereo camera system using the first stereo image sequences.

	Camera 1	Camera 2
Feature detection and tracking (second per frame)	0.127s	0.131s
Stereo correspondence matching (second per frame)	0.200s	
Camera pose tracking (second per frame)	EKF-2: 0.043s UKF-2: 0.176s DE-2: 69.850s	
All the algorithms were implemented in Matlab 7.4 and run on a computer with Pentium D 3.0 GHz CPU and 1 GB RAM.		

Table 3.7: Timings of the real experiment of pose tracking of the stereo camera system using the second stereo image sequences.

	Camera 1	Camera 2
Feature detection and tracking (second per frame)	0.189s	0.184s
Stereo correspondence matching (second per frame)	0.174s	
Camera pose tracking (second per frame)	EKF-2: 0.020s UKF-2: 0.089s DE-2: 59.057s	
All the algorithms were implemented in Matlab 7.4 and run on a computer with Pentium D 3.0 GHz CPU and 1 GB RAM.		

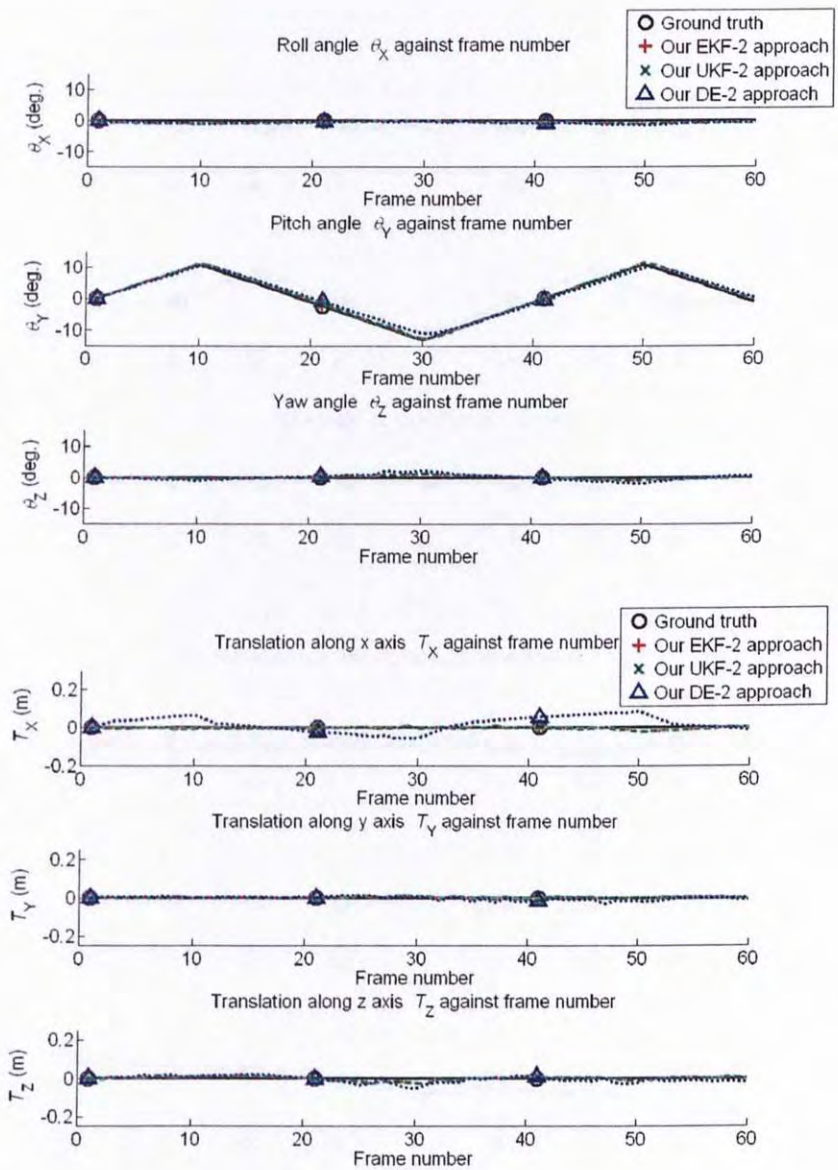

Figure 3.16: Result of the real experiment of pose tracking of the stereo camera system using the second stereo image sequences. (Top) Rotation. (Bottom) Translation.

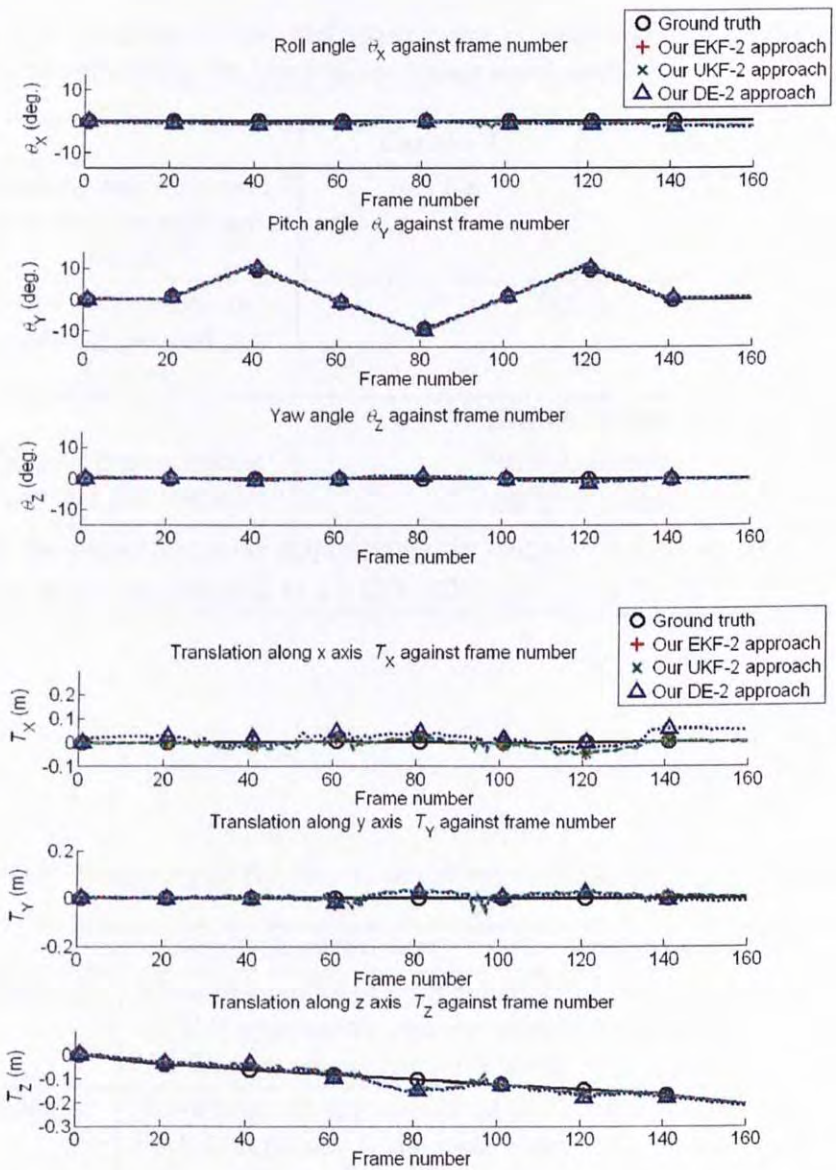

Figure 3.17: Result of the real experiment of pose tracking of the stereo camera system using the third stereo image sequences. (Top) Rotation. (Bottom) Translation.

Table 3.8: Timings of the real experiment of pose tracking of the stereo camera system using the third stereo image sequences.

	Camera 1	Camera 2
Feature detection and tracking (second per frame)	0.174s	0.172s
Stereo correspondence matching (second per frame)	0.222s	
Camera pose tracking (second per frame)	EKF-2: 0.023s UKF-2: 0.089s DE-2: 55.186s	
All the algorithms were implemented in Matlab 7.4 and run on a computer with Pentium D 3.0 GHz CPU and 1 GB RAM.		

Table 3.9: Summary of the results of the synthetic and real experiments.

	Synthetic experiment	Real experiment
Accuracy:	From the results of both experiments, our EKF-2 and UKF-2 approaches recover accurate poses while our DE-2 approach recovers less accurate poses.	
Efficiency:	According to the results of both experiments, our EKF-2 approach is the most efficient while our DE-2 approach is the least efficient.	
Conclusion:	The results of the synthetic and real experiment are consistent.	

3.5 Summary

In this chapter, we proposed an algorithm based on the modelless scheme to estimate the orientation and location of a stereo camera system. Our algorithm can recover more accurate pose than the existing algorithms in terms of accuracy by using the Rodrigues' formula. It can also be efficient enough to work in real-time as our algorithm makes use of the trifocal tensor constraints to bypass the computation of the structure. However, the 3-D model can still be reconstructed if the application requires.

In addition, performances of different estimation methods including the unscented Kalman filter, extended Kalman filter, and differential evolution used in our approach were compared and analyzed. The UKF-based approach is expected to handle a non-linear system more robustly than the EKF-based approach. However, the synthetic data experiment demonstrated that the EKF-based approach and the UKF-based approach have similar performances in terms of accuracy for our problem. It is believed that the assumption of local linearity in the extended Kalman filter is appropriate in our problem. We also found that the differential evolution is not suitable for our problem as shown in the experiments. It is slow and not very accurate. However, it can be further investigated when there is parallel computing.

□ **End of chapter.**

Chapter 4

Advance to two pairs of stereo cameras

4.1 Overview

The previous chapter focuses on pose tracking of a pair of stereo camera. In this chapter, we focus on pose tracking of a multiple camera system consisting of two pairs of stereo cameras. The multiple camera system can have the advantages of larger field of view. Our system can work even when one stereo pair does not contain enough features.

4.1.1 Related work

There is some research work related to multiple camera systems [20] [25] [3]. A multiple camera system consisting of six cameras is treated as a single camera in [3]. Calibration and pose tracking methods for this system were proposed. In other research, multiple camera systems are used for computing the pose of a moving object [20] and visual servoing [25].

4.1.2 Contribution

In this chapter, we propose an algorithm to track the pose of a multiple camera system consisting of two pairs of stereo cameras.

- The proposed algorithm can recover the orientation and location of the multiple camera system accurately. As the system consists of two pairs of stereo cameras, it can provide more features to track the pose more accurately when compared with the approach proposed in the previous chapter. Our approach can work even when one stereo pair is blocked.
- Our algorithm is efficient. There is no explicit computation of the structure and the dimension of the state used in the Kalman filters is six, which is minimal. These characteristics enable our algorithm to work in real-time.
- Performances of different estimation methods including the extended Kalman filter, which handles a non-linear system by assuming local linearity, and the unscented Kalman filter, which handles a non-linear system by statistical calculations, used in our approach are compared and analyzed. Their advantages and disadvantages are discussed throughout the experiment.
- Different orientations between the two stereo pairs are studied to investigate their effects on the accuracy of the recovered pose.

4.2 Problem definition

The geometry model is illustrated in figure 4.1. The projection of the mth 3-D model point $\mathbf{X}_m^O$ (homogeneous coordinate) on the image plane of camera i is $\mathbf{u}_{m,t}^{Ci}$ (homogeneous coordinate) at time t. The relationships between the 3-D model points and their projections on the image plane of all the cameras are shown in equation (4.1).

$$\begin{aligned}
\mathbf{u}_{m,t}^{C1} &= \mathbf{K}\begin{bmatrix}\mathbf{I}_{3\times3} & \mathbf{0}_{3\times1}\end{bmatrix}\mathbf{M}_t\mathbf{X}_m^O \\
\mathbf{u}_{m,t}^{C2} &= \mathbf{K}\begin{bmatrix}\mathbf{I}_{3\times3} & \mathbf{0}_{3\times1}\end{bmatrix}\mathbf{B}_{12}\mathbf{M}_t\mathbf{X}_m^O \\
\mathbf{u}_{m,t}^{C3} &= \mathbf{K}\begin{bmatrix}\mathbf{I}_{3\times3} & \mathbf{0}_{3\times1}\end{bmatrix}\mathbf{B}_{13}\mathbf{M}_t\mathbf{X}_m^O \\
\mathbf{u}_{m,t}^{C4} &= \mathbf{K}\begin{bmatrix}\mathbf{I}_{3\times3} & \mathbf{0}_{3\times1}\end{bmatrix}\mathbf{B}_{34}\mathbf{B}_{13}\mathbf{M}_t\mathbf{X}_m^O
\end{aligned} \tag{4.1}$$

The object coordinate frame acts as the reference coordinate frame which is the same as the camera 1 coordinate frame at time 0. $\mathbf{M}_t$ is a 4×4 matrix that transforms 3-D points from the object (reference) coordinate frame to camera 1 coordinate frame at time t. There are two pairs of stereo cameras not necessarily placed back-to-back. Camera 1 and camera 2 form a stereo pair while camera 3 and camera 4 form another. $\mathbf{B}_{12}$, $\mathbf{B}_{13}$, $\mathbf{B}_{34}$ are 4×4 matrices that represent the rigid transformations from the coordinate frame of camera 1 to that of camera 2, from the coordinate frame of camera 1 to that of camera 3, and from the coordinate frame of camera 3 to that of camera 4 respectively. $\mathbf{K}$ is a 3×3 matrix that encapsulates the intrinsic parameters (including focal length, image center, and pixel size) of the cameras and is introduced in equation (2.3). For the sake

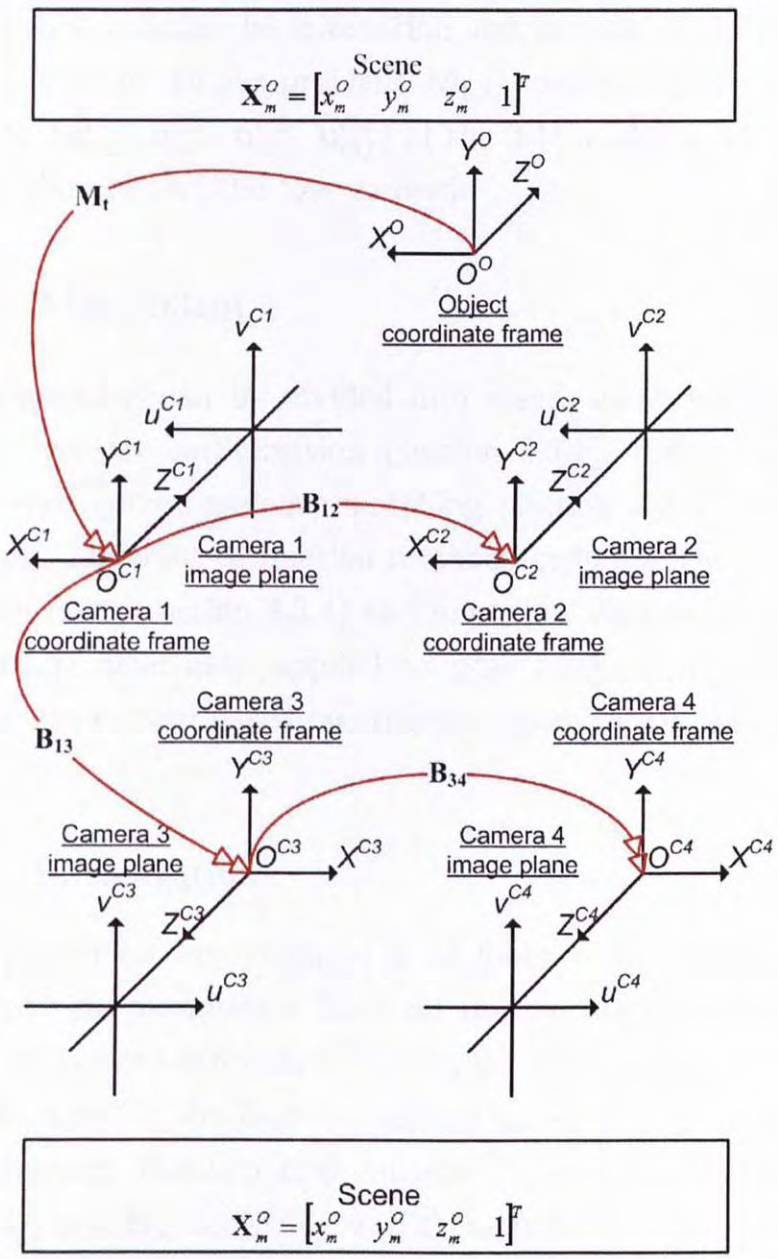

Figure 4.1: The image formation model of two pairs of stereo cameras.

of clarity, all the cameras are assumed to have the same intrinsic parameters.

$\mathbf{M}_t$ encapsulates the orientation and location of the multiple camera system. In our problem, $\mathbf{M}_t$ is recovered given the projections ($\mathbf{u}_{m,t}^{C1}$, $\mathbf{u}_{m,t}^{C2}$, $\mathbf{u}_{m,t}^{C3}$, $\mathbf{u}_{m,t}^{C4}$) of the 3-D model points on the image planes of all the four cameras.

4.3 Algorithm

The algorithm can be divided into stages as shown in figure 4.2. They are initialization (section 4.3.1), feature tracking and stereo correspondence matching (section 4.3.2), and pose tracking. Different estimation methods including the extended Kalman filter (section 4.3.4) and unscented Kalman filter (section 4.3.5) have been applied to pose tracking of the multiple camera system based on trifocal tensor constraints (section 4.3.3).

4.3.1 Initialization

Some parameters are required to be found in the initialization. The intrinsic parameters $\mathbf{K}$ of all the cameras are calibrated using the camera calibration toolbox [5]. The matrices $\mathbf{B}_{12}$, $\mathbf{B}_{13}$, and $\mathbf{B}_{34}$ can be obtained by calibration [4] [12] [3] or manual measurement. Fundamental matrices $\mathbf{F}_{12}$ and $\mathbf{F}_{34}$ are obtained from $\mathbf{B}_{12}$ and $\mathbf{B}_{34}$ according to [13] respectively. All parameters found in the initialization would remain unchanged in the latter frames.

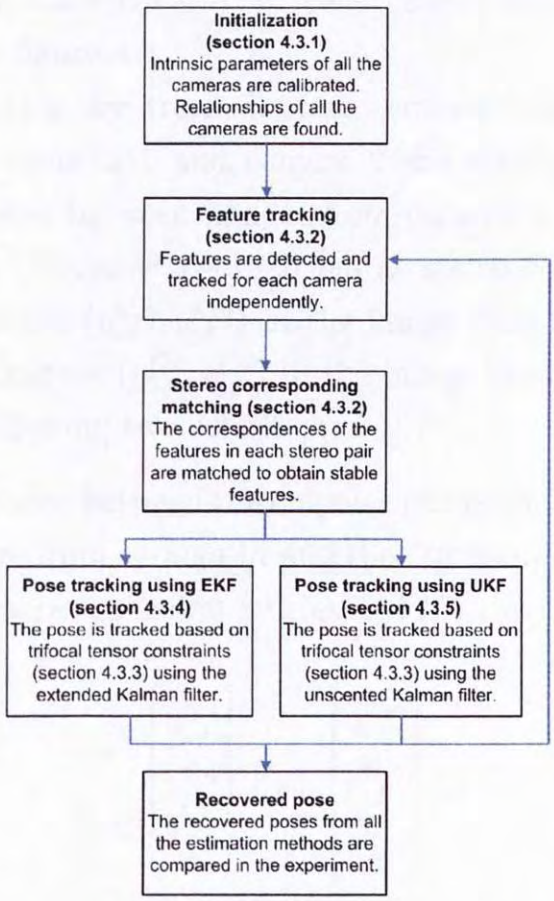

Figure 4.2: The overall algorithm for pose tracking of two pairs of stereo cameras.

4.3.2 Feature tracking and stereo correspondence matching

Features are detected and tracked from the image sequences using the Kanade-Lucas-Tomasi feature tracker [30] at each time step. Features are tracked for each camera independently as illustrated in figure 4.3.

After features are tracked, stereo correspondences between images from camera 1 and camera 2 are matched and stereo correspondences between images from camera 3 and camera 4 are matched. Features are matched as stereo correspondences if the ith feature $(u_{i,t}^{Cm}, v_{i,t}^{Cm})$ in the image from the camera m and the jth feature $(u_{j,t}^{Cn}, v_{j,t}^{Cn})$ in the image from the camera n satisfy the following two conditions.

1. The distance between the epipolar line of the ith features in the image from camera m and the jth feature in the image from camera n is below a threshold D as shown in equation (4.2).

$$\begin{bmatrix} u_{j,t}^{Cn} \\ v_{j,t}^{Cn} \\ 1 \end{bmatrix} \mathbf{F_{mn}} \begin{bmatrix} u_{i,t}^{Cm} \\ v_{i,t}^{Cm} \\ 1 \end{bmatrix} \leq D \tag{4.2}$$

2. The templates of the ith feature in the image from camera m and the jth feature in the image from camera n have large normalized cross-correlation value.

where $m = 1$ and $n = 2$ are for the stereo pair consisting of camera 1 and camera 2 while $m = 3$ and $n = 4$ are for the stereo pair consisting of camera 3 and camera 4.

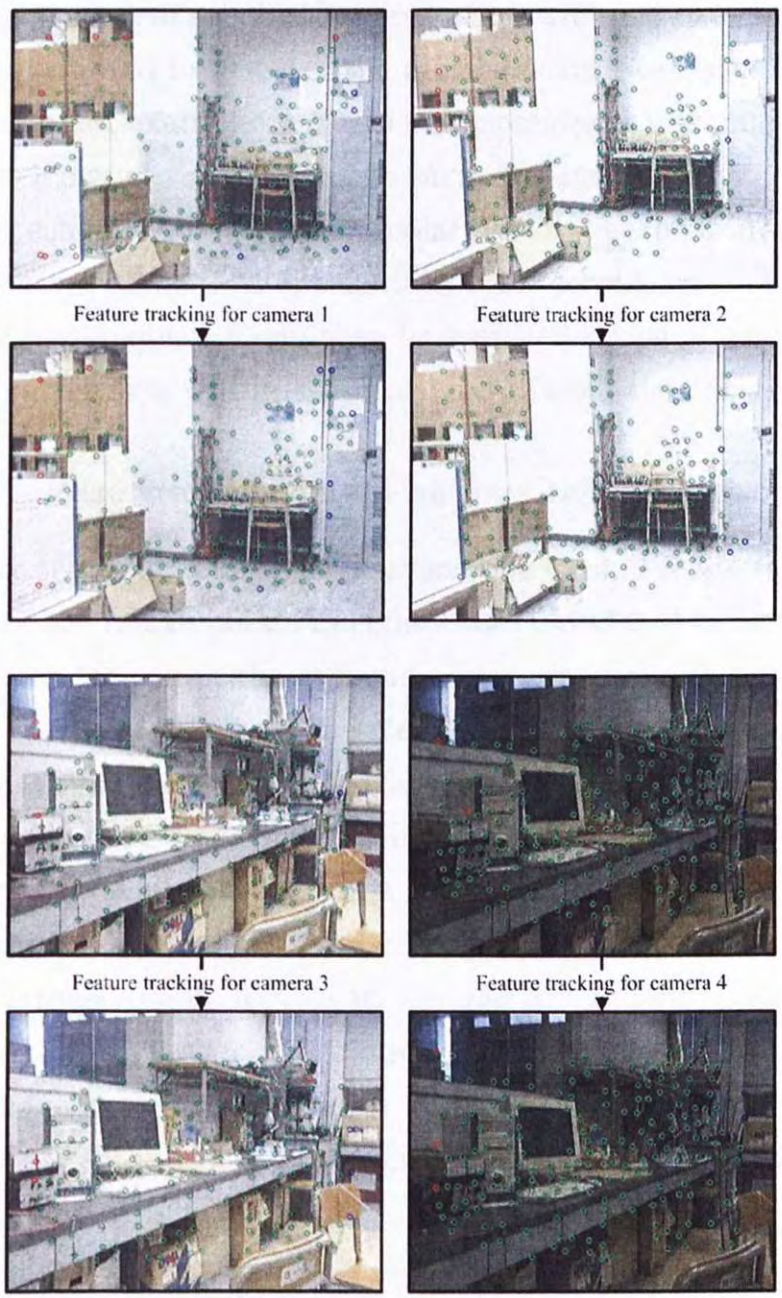

Figure 4.3: Illustration of feature tracking in pose tracking of two pairs of stereo cameras.

Features without correspondence are rejected as outliers to maintain a set of reliable features which are used to recover the orientation and location of the multiple camera system. Figure 4.4 shows an example of stereo correspondence matching.

Efficiency of the process can be maintained by using the following scheme. Firstly, the epipolar line of the ith feature in the image from camera m is calculated. Its correspondence in the image from camera n can then be searched by using normalized cross correlation within the threshold D from the epipolar line.

4.3.3 Pose tracking based on four trifocal tensors

In pose tracking, four trifocal tensors are used to relate the positions of the image points in eight views illustrated in figure 4.5. One time step from the sequences is considered as a base time step. Initially, the first time step (i.e. time 0) is considered as the base time step. Both trifocal tensor 1 $\mathbf{T}^1$ and trifocal tensor 2 $\mathbf{T}^2$ take the stereo image pair by camera 1 and camera 2 at the base time step as their first two views. The image captured by camera 1 at time t is considered as the third view for tensor 1 $\mathbf{T}^1$. Matched points $\mathbf{u}^{C1}_{m,base}$, $\mathbf{u}^{C2}_{m,base}$ and $\mathbf{u}^{C1}_{m,t}$ in these three views are related by tensor 1 $\mathbf{T}^1$, where $m \in (1, \ldots, N_{12})$. Similarly, the image captured by camera 2 at time t is considered as the third view for tensor 2 $\mathbf{T}^2$, which relates matched points $\mathbf{u}^{C1}_{m,base}$, $\mathbf{u}^{C2}_{m,base}$ and $\mathbf{u}^{C2}_{m,t}$, where $m \in (1, \ldots, N_{12})$.

Both trifocal tensor 3 $\mathbf{T}^3$ and trifocal tensor 4 $\mathbf{T}^4$ take the stereo image pair by camera 3 and camera 4 at the base time step as their first two views. The image captured by camera 3 at

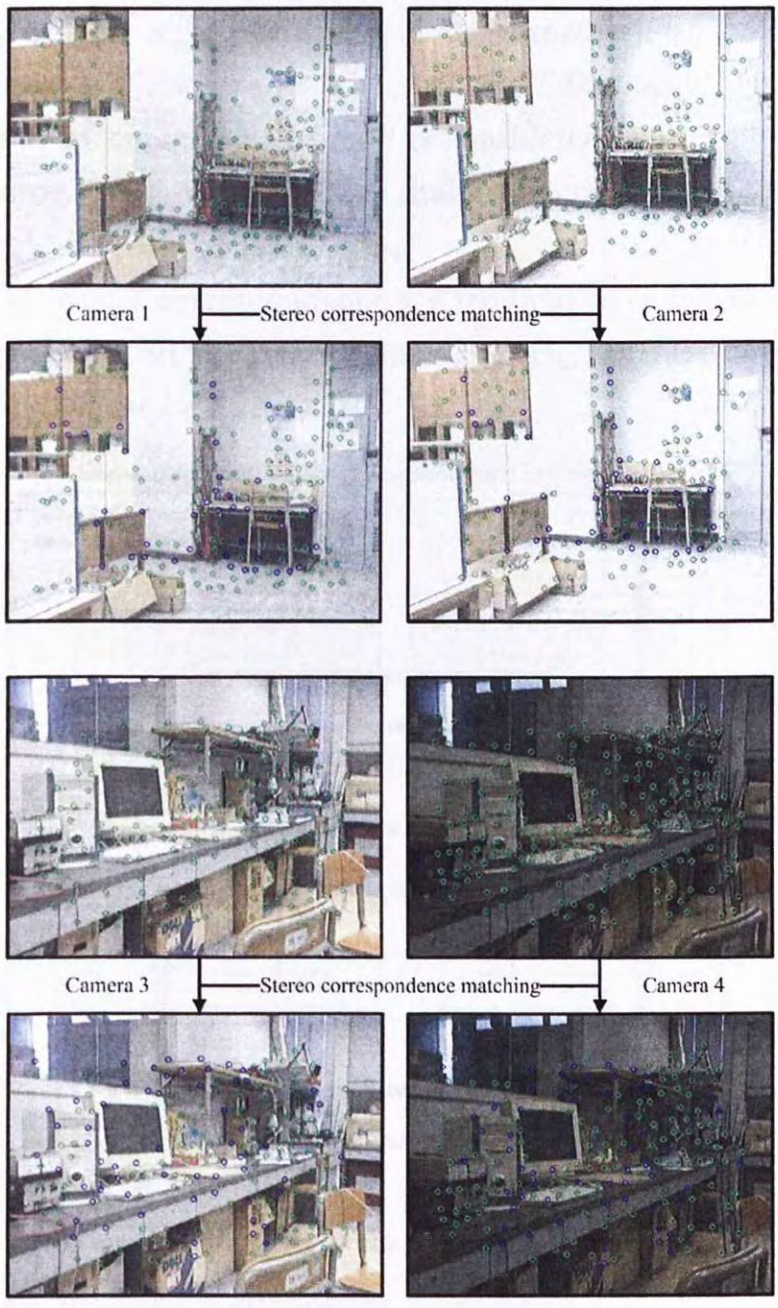

Figure 4.4: Illustration of stereo correspondence matching in pose tracking of two pairs of stereo cameras.

time t is considered as the third view for tensor 3 $\mathbf{T}^3$. Matched points $\mathbf{u}^{C3}_{m,base}$, $\mathbf{u}^{C4}_{m,base}$ and $\mathbf{u}^{C3}_{m,t}$ in these three views are related by tensor 3 $\mathbf{T}^3$, where $m \in (1, \ldots, N_{34})$. Similarly, the image captured by camera 4 at time t is considered as the third view for tensor 4 $\mathbf{T}^4$, which relates matched points $\mathbf{u}^{C3}_{m,base}$, $\mathbf{u}^{C4}_{m,base}$ and $\mathbf{u}^{C4}_{m,t}$, where $m \in (1, \ldots, N_{34})$.

As at least 7 correspondence are required to calculate a trifocal tensor and all the relationships $\mathbf{B}_{12}$, $\mathbf{B}_{13}$, and $\mathbf{B}_{34}$ are fixed, $N_{12} + N_{34}$ must be at least 7.

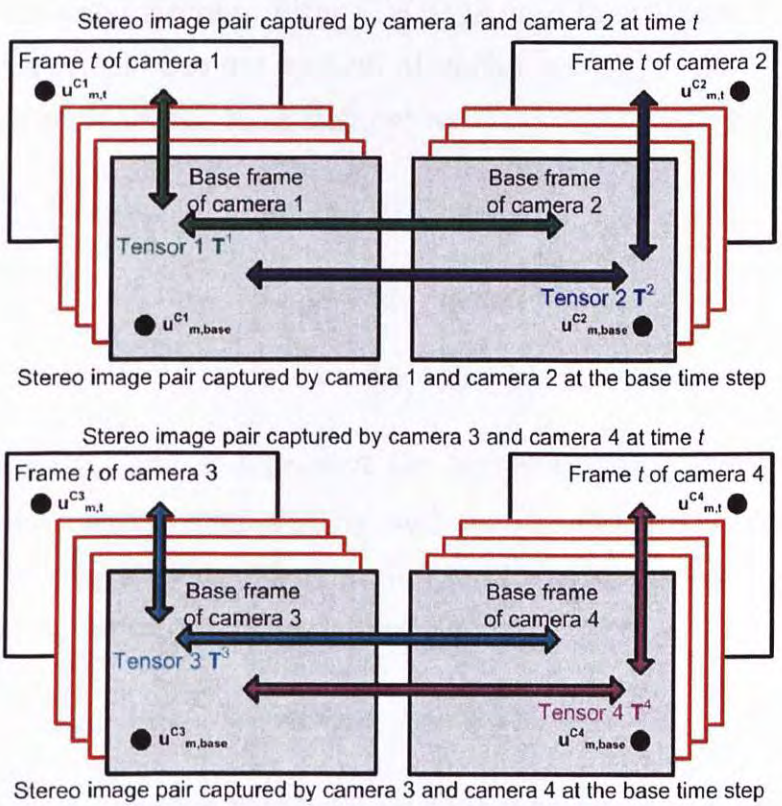

Figure 4.5: Illustration of the use of four trifocal tensors in pose tracking of two pairs of stereo cameras.

However, when the multiple camera system moves, features

may disappear and new features may appear. The number of correspondences $N_{12}+N_{34}$ may be less than 7. In this situation, the base time step needs to be reset. The time step $t-1$ become a new base time step. For the sake of clarity, we assume that there are no changes of the base time step required in section 4.3.4 and section 4.3.5.

4.3.4 Pose tracking using extended Kalman filter (Our EKF-4 approach)

The extended Kalman filter (EKF) is used to estimate the state of the multiple camera system at each time step.

The *state* vector $\mathbf{x}_t$ is defined as

$$\mathbf{x}_t = \begin{bmatrix} \dot{\mathbf{T}}_t^T & \dot{\mathbf{w}}_t^T \end{bmatrix}^T \tag{4.3}$$

$$\dot{\mathbf{T}}_t = \begin{bmatrix} \dot{x}_t & \dot{y}_t & \dot{z}_t \end{bmatrix}^T \tag{4.4}$$

$$\dot{\mathbf{w}}_t = \begin{bmatrix} \dot{\alpha}_t & \dot{\beta}_t & \dot{\gamma}_t \end{bmatrix}^T \tag{4.5}$$

where $\dot{x}_t$, $\dot{y}_t$, and $\dot{z}_t$ represent the translational velocities along x, y, and z-axes respectively and $\dot{\alpha}_t$, $\dot{\beta}_t$, and $\dot{\gamma}_t$ represent the angular velocities about x, y, and z-axes respectively.

The *dynamic model* is defined as

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \mathbf{v} \tag{4.6}$$

$\mathbf{v}$ is a 6×1 vector representing Gaussian process noise which models the changes of the velocities of the multiple camera system. The pose $\mathbf{M}_t$ of the multiple camera system is computed

using

$$\mathbf{M}_t = \begin{bmatrix} e^{\tilde{\dot{\mathbf{w}}}_t} & \dot{\mathbf{T}}_t \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \mathbf{M}_{t-1} \tag{4.7}$$

$$= \begin{bmatrix} e^{\tilde{\dot{\mathbf{w}}}_t} & \dot{\mathbf{T}}_t \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{t-1} & \mathbf{T}_{t-1} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{4.8}$$

$$\tag{4.9}$$

where

$$\tilde{\dot{\mathbf{w}}}_t = \begin{bmatrix} 0 & -\dot{\gamma}_t & \dot{\beta}_t \\ \dot{\gamma}_t & 0 & -\dot{\alpha}_t \\ -\dot{\beta}_t & \dot{\alpha}_t & 0 \end{bmatrix} \tag{4.10}$$

The exponential of the skew-symmetric matrix $\tilde{\dot{\mathbf{w}}}_t$ can be calculated using the Rodrigues' formula (4.11).

$$e^{\tilde{\dot{\mathbf{w}}}_t} = \mathbf{I} + \frac{e^{\tilde{\dot{\mathbf{w}}}_t}}{\|e^{\tilde{\dot{\mathbf{w}}}_t}\|} \sin \|e^{\tilde{\dot{\mathbf{w}}}_t}\| + \frac{e^{\tilde{\dot{\mathbf{w}}}_t}{}^2}{\|e^{\tilde{\dot{\mathbf{w}}}_t}\|^2} (1 - \cos \|e^{\tilde{\dot{\mathbf{w}}}_t}\|) \tag{4.11}$$

The *measurement model* is defined as

$$\mathbf{y}_t = \mathbf{h}_t(\mathbf{x}_t) + \mathbf{n} = \mathbf{g}_t(\mathbf{M}_t) + \mathbf{n} \tag{4.12}$$

where $\mathbf{n}$ is a $4(N_{12} + N_{34}) \times 1$ vector representing Gaussian measurement noise (N_{12} is the number of available correspondences in the stereo image pair by camera 1 and camera 2 and N_{34} is the number of available correspondences in the stereo image pair by the camera 3 and camera 4). $\mathbf{g}_t(\mathbf{M}_t)$ is a function to compute

the projections of all the available feature points on the image planes of camera 1, camera 2, camera 3, and camera 4 at time t as shown in equation (4.13).

$$\begin{aligned}\mathbf{g}_t(\mathbf{M}_t) &= \left(\begin{pmatrix} u_{1,t}^{C1} \\ v_{1,t}^{C1} \\ \vdots \\ u_{m,t}^{C1} \\ v_{m,t}^{C1} \\ \vdots \\ u_{N_{12},t}^{C1} \\ v_{N_{12},t}^{C1} \end{pmatrix}^T \begin{pmatrix} u_{1,t}^{C2} \\ v_{1,t}^{C2} \\ \vdots \\ u_{m,t}^{C2} \\ v_{m,t}^{C2} \\ \vdots \\ u_{N_{12},t}^{C2} \\ v_{N_{12},t}^{C2} \end{pmatrix}^T \begin{pmatrix} u_{1,t}^{C3} \\ v_{1,t}^{C3} \\ \vdots \\ u_{n,t}^{C3} \\ v_{n,t}^{C3} \\ \vdots \\ u_{N_{34},t}^{C3} \\ v_{N_{34},t}^{C3} \end{pmatrix}^T \begin{pmatrix} u_{1,t}^{C4} \\ v_{1,t}^{C4} \\ \vdots \\ u_{n,t}^{C4} \\ v_{n,t}^{C4} \\ \vdots \\ u_{N_{34},t}^{C4} \\ v_{N_{34},t}^{C4} \end{pmatrix}^T \right)^T \\ &= \left(\mathbf{g}_{1t}(\mathbf{M}_t)^T \quad \mathbf{g}_{2t}(\mathbf{M}_t)^T\right)^T \end{aligned} \tag{4.13}$$

where $(u_{m,t}^{C1}, v_{m,t}^{C1})$ and $(u_{m,t}^{C2}, v_{m,t}^{C2})$ are the stereo correspondences in the stereo image pair by camera 1 and camera 2 at time t and $(u_{n,t}^{C3}, v_{n,t}^{C3})$ and $(u_{n,t}^{C4}, v_{n,t}^{C4})$ are the stereo correspondences in the stereo image pair by camera 3 and camera 4 at time t. In our approach, we consider $\mathbf{g}_{1t}(\mathbf{M}_t)$ to be a function to compute the projections of all the available feature points on the image planes of camera 1 and camera 2 at time t and $\mathbf{g}_{2t}(\mathbf{M}_t)$ to be a function to compute the projections of all the available feature points on the image planes of camera 3 and camera 4 at time t.

Based on the point transfer using $\mathbf{T}^1$ and $\mathbf{T}^2$ illustrated in figure 4.5, $\mathbf{g}_{1t}(\mathbf{M}_t)$ is calculated using equation (4.14) and equa-

tion (4.15) represented in tensor notation. Details can be found in [13].

$$(U_{m,t}^{C1})^k = (U_{m,base}^{C1})^i (l_{m,base}^{C2})_j (T^1)_i^{jk} \tag{4.14}$$

$$(U_{m,t}^{C2})^k = (U_{m,base}^{C1})^i (l_{m,base}^{C2})_j (T^2)_i^{jk} \tag{4.15}$$

$\mathbf{U}_{m,t}^{Ci}$ is computed according to equation (4.16) to remove the effects of the intrinsic parameters of the cameras.

$$\mathbf{U}_{m,t}^{Ci} = \begin{pmatrix} (U_{m,t}^{Ci})^1 \\ (U_{m,t}^{Ci})^2 \\ (U_{m,t}^{Ci})^3 \end{pmatrix} = \mathbf{K}^{-1} \begin{pmatrix} u_{m,t}^{Ci} \\ v_{m,t}^{Ci} \\ 1 \end{pmatrix} \tag{4.16}$$

$\mathbf{l}_{m,base}^{C2}$ is a line passing through the mth feature point on the image plane of camera 2 at the base time step and can be found according to equation (4.17).

$$\mathbf{l}_{m,base}^{C2} = \begin{pmatrix} (l_{m,base}^{C2})_1 \\ (l_{m,base}^{C2})_2 \\ (l_{m,base}^{C2})_3 \end{pmatrix} = \begin{pmatrix} (l_{m,base})_2 \\ -(l_{m,base})_1 \\ -(U_{m,t}^{C2})^1 (l_{m,base})_2 + (U_{m,t}^{C2})^2 (l_{m,base})_1 \end{pmatrix}$$

$$\mathbf{l}_{m,base_e} = \begin{pmatrix} (l_{m,base})_1 \\ (l_{m,base})_2 \\ (l_{m,base})_3 \end{pmatrix} = \mathbf{e}_{12} \times \mathbf{U}_{m,base}^{C2} \tag{4.17}$$

where $\mathbf{e}_{12}$ is the epipole observed from camera 2 and $\mathbf{l}_{m,base_e}$ is the epipolar line passing through the mth feature point on the image plane of camera 2 at the base time step.

$(T^1)_i^{jk}$ and $(T^2)_i^{jk}$ in equation (3.13) and equation (3.14) represent the elements at the position (i, j, k) of $\mathbf{T}^1$ and $\mathbf{T}^2$ respectively. $\mathbf{T}^1$ and $\mathbf{T}^2$ are the two trifocal tensors illustrated in figure 3.5. Consider that $[\mathbf{I}_{3\times3}\ \mathbf{0}_{3\times1}]$ and $[\mathbf{I}_{3\times3}\ \mathbf{0}_{3\times1}]\mathbf{B}_{12} = [b_i^j]$ are the extrinsic parameters of camera 1 and camera 2 at the base time step respectively and $[\mathbf{I}_{3\times3}\ \mathbf{0}_{3\times1}]\mathbf{M}_t = [a_i^j]$ and $[\mathbf{I}_{3\times3}\ \mathbf{0}_{3\times1}]\mathbf{B}_{12}\mathbf{M}_t = [a_i'^j]$) are the extrinsic parameters of camera 1 and camera 2 at time t respectively, $\mathbf{T}^1$ and $\mathbf{T}^2$ can be computed using equation (4.18).

$$\begin{aligned} T^{1jk}_{\ i} &= a_i^j b_4^k - a_4^j b_i^k \\ T^{2jk}_{\ i} &= a_i'^j b_4^k - a_4'^j b_i^k \end{aligned} \tag{4.18}$$

$\mathbf{g}_{2t}(\mathbf{M}_t)$ is the measurement function for the stereo pair consisting of camera 3 and camera 4. It is the point transfer using $\mathbf{T}^3$ and $\mathbf{T}^4$ illustrated in figure 4.5. $\mathbf{g}_{2t}(\mathbf{M}_t)$ can be computed using the function $\mathbf{g}_{1t}$ by calculating the pose of the stereo pair consisting of camera 3 and camera 4 according to equation (4.19).

$$\mathbf{g}_{2t}(\mathbf{M}_t) = \mathbf{g}_{1t}(\mathbf{B}_{13}\mathbf{M}_t{\mathbf{B}_{13}}^{-1}) \tag{4.19}$$

With the state, dynamic model, and measurement model defined, equations required for the EKF are derived according to [11] as follows.

Time update equations are

$$\begin{aligned} \hat{\mathbf{x}}_t^- &= \hat{\mathbf{x}}_{t-1} \\ \mathbf{P}_t^- &= \mathbf{P}_{t-1} + \mathbf{R}^{\mathbf{v}} \end{aligned} \tag{4.20}$$

Measurement update equations are

$$\begin{aligned}
\mathbf{K}_t &= \mathbf{P}_t^- \nabla \mathbf{h}_t|_{\hat{\mathbf{x}}_t^-}{}^T (\nabla \mathbf{h}_t|_{\hat{\mathbf{x}}_t^-} \mathbf{P}_t^- \nabla \mathbf{h}_t|_{\hat{\mathbf{x}}_t^-}{}^T + \mathbf{R}^{\mathbf{n}})^{-1} \\
\hat{\mathbf{x}}_t &= \hat{\mathbf{x}}_t^- + \mathbf{K}_t(\mathbf{y}_t - \mathbf{h}_t(\hat{\mathbf{x}}_t^-)) \\
\mathbf{P}_t &= (\mathbf{I} - \mathbf{K}_t \nabla \mathbf{h}_t|_{\hat{\mathbf{x}}_t^-}) \mathbf{P}_t^-
\end{aligned} \tag{4.21}$$

$\hat{\mathbf{x}}_t^-$ and $\hat{\mathbf{x}}_t$ are the states after time update and measurement update respectively. $\mathbf{P}_t^-$ and $\mathbf{P}_t$ are the 6×6 covariance matrices of the state $\hat{\mathbf{x}}_t^-$ and $\hat{\mathbf{x}}_t$ respectively. $\mathbf{R}^{\mathbf{v}}$ and $\mathbf{R}^{\mathbf{n}}$ are the 6×6 covariance matrices of the process noise $\mathbf{v}$ and the measurement noise $\mathbf{n}$ respectively. $\nabla \mathbf{h}_t|_{\hat{\mathbf{x}}_t^-}$ is the Jacobian of the measurement equation $\mathbf{h}_t(\mathbf{x})$ at $\hat{\mathbf{x}}_t^-$. $\mathbf{K}_t$ is the $6 \times 4(N_{12} + N_{34})$ Kalman gain matrix in the extended Kalman filter.

4.3.5 Pose tracking using unscented Kalman filter (Our UKF-4 approach)

The unscented Kalman filter (UKF) is used to estimate the velocity of the multiple camera system at each time step. The definitions of the state, dynamic model, and measurement model used in the UKF are the same as those used in the EKF. They are described in section 4.3.4. With the state, dynamic model, and measurement model defined, equations required for the UKF are derived according to [34].

Time update equations are

$$\begin{aligned}
\hat{\mathbf{x}}_t^- &= \hat{\mathbf{x}}_{t-1} \\
\mathbf{P}_t^- &= \mathbf{P}_{t-1} + \mathbf{R}^{\mathbf{v}} \\
\mathbf{X}_{t|t-1} &= \left[\hat{\mathbf{x}}_t^- \quad \hat{\mathbf{x}}_t^- + \sqrt{(L+\lambda)\mathbf{P}_t^-} \quad \hat{\mathbf{x}}_t^- - \sqrt{(L+\lambda)\mathbf{P}_t^-}\right] \\
\mathbf{Y}_{t|t-1} &= \mathbf{h}_t^*(\mathbf{X}_{t|t-1}) \\
\hat{\mathbf{y}}_t^- &= \sum_{i=0}^{2L} W_i^{(m)} Y_{i,t|t-1}
\end{aligned} \tag{4.22}$$

Measurement update equations are

$$\begin{aligned}
\mathbf{P}_{\tilde{\mathbf{y}}_t\tilde{\mathbf{y}}_t} &= \sum_{i=0}^{2L} W_i^{(c)} (Y_{i,t|t-1} - \hat{\mathbf{y}}_t^-)(Y_{i,t|t-1} - \hat{\mathbf{y}}_t^-)^T + \mathbf{R}^{\mathbf{n}} \\
\mathbf{P}_{\mathbf{x}_t\mathbf{y}_t} &= \sum_{i=0}^{2L} W_i^{(c)} (X_{i,t|t-1} - \hat{\mathbf{x}}_t^-)(Y_{i,t|t-1} - \hat{\mathbf{y}}_t^-)^T \\
\mathbf{K}_t &= \mathbf{P}_{\mathbf{x}_t\mathbf{y}_t} \mathbf{P}_{\tilde{\mathbf{y}}_t\tilde{\mathbf{y}}_t}{}^{-1} \\
\hat{\mathbf{x}}_t &= \hat{\mathbf{x}}_t^- + \mathbf{K}_t(\mathbf{y}_t - \hat{\mathbf{y}}_t^-) \\
\mathbf{P}_t &= \mathbf{P}_t^- - \mathbf{K}_t \mathbf{P}_{\tilde{\mathbf{y}}_t\tilde{\mathbf{y}}_t} \mathbf{K}_t{}^T
\end{aligned} \tag{4.23}$$

where

$$\begin{aligned}
W_0^{(m)} &= \frac{\lambda}{L+\lambda} \\
W_0^{(c)} &= \frac{\lambda}{L+\lambda} + 1 - \alpha^2 + \beta \\
W_i^{(m)} &= W_i^{(c)} = \frac{1}{2(L+\lambda)}, i = 1, 2, \ldots, 2L
\end{aligned} \tag{4.24}$$

λ is a scaling parameter. α and β represent the spread of the sigma points and the prior knowledge of the distribution of the

state $\mathbf{x}_t$ respectively. $\mathbf{X}_{t|t-1}$ contains all the sigma points used in the unscented transform while $\mathbf{X}_{i,t|t-1}$ indicates the ith sigma point. $L = 6$ is the dimension of the state $\mathbf{x}_t$. $W_0^{(m)}$ and $W_0^{(c)}$ are the weights used in calculating the mean and the covariance matrices respectively. $\mathbf{h}_t^*(\mathbf{X}_{t|t-1})$ is a function which calculates all $2L+1$ sigma points in $\mathbf{X}_{t|t-1}$ using $\mathbf{h}_t(\mathbf{x}_t)$ defined in equation (4.12). That is

$$\begin{aligned} &\mathbf{h}_t^*(\mathbf{X}_{t|t-1}) = \mathbf{Y}_{t|t-1} \\ &\text{means} \\ &\mathbf{h}_t(\mathbf{X}_{i,t|t-1}) = \mathbf{Y}_{i,t|t-1} \\ &i \in (1, 2, \ldots, 2L, 2L+1) \end{aligned} \tag{4.25}$$

$\hat{\mathbf{y}}_t^-$ is the predicted measurement computed from the unscented transform at time t. $\mathbf{P}_{\tilde{\mathbf{y}}_t\tilde{\mathbf{y}}_t}$ is the $4(N_{12}+N_{34}) \times 4(N_{12}+N_{34})$ covariance matrix between the elements of the predicted measurement $\hat{\mathbf{y}}_t^-$. $\mathbf{P}_{\mathbf{x}_t\mathbf{y}_t}$ is the $6 \times 4(N_{12}+N_{34})$ covariance matrix between the elements of the state $\hat{\mathbf{x}}_t^-$ and the predicted measurement $\hat{\mathbf{y}}_t^-$. Similar to the EKF, $\mathbf{R}^{\mathbf{v}}$ and $\mathbf{R}^{\mathbf{n}}$ are the 6×6 covariance matrices of the process noise $\mathbf{v}$ and the measurement noise $\mathbf{n}$ respectively. $\hat{\mathbf{x}}_t^-$ and $\hat{\mathbf{x}}_t$ are the states at time t after time update and measurement update respectively. $\mathbf{P}_t^-$ and $\mathbf{P}_t$ are the 6×6 covariance matrices of the states $\hat{\mathbf{x}}_t^-$ and $\hat{\mathbf{x}}_t$ respectively. $\mathbf{K}_t$ is the $6 \times 4(N_{12}+N_{34})$ Kalman gain matrix at time t. The mean and covariance are propagated using the unscented transform in the UKF.

4.4 Experiment

4.4.1 Synthetic experiments

Five approaches were tested in the experiment using synthetic data. These five approaches included our EKF-4 approach and UKF-4 approach for pose tracking of two pairs of stereo cameras. These two approaches were compared with the approach proposed by Yu [38] and our EKF-2 approach and UKF-2 approach for pose tracking of a stereo camera system described in chapter 3. The details are summarized in table 4.1. All the approaches were implemented in Matlab 7.4 and run on a computer with Pentium D 3.0 GHz CPU and 1 GB RAM. The objective of the experiment is to compare the performances of these five methods in terms of accuracy and efficiency.

In the synthetic data experiment, all the four cameras with resolution 640×480 had 4.6 mm focal length and a 2-D zero-mean Gaussian noise of 1 pixel standard deviation. Cameras in both the two stereo pairs were put 0.1 m apart while the two pairs were placed back-to-back with 0.1 m apart. The setting of the cameras is illustrated in figure 4.6. Five hundred model points were generated randomly in 3-D space at places $1 - 6$ m from camera 1. Each test sequence consisted of 90 frames. The motion of the multiple camera system consisted of three different segments which included mixed motion (both rotation and translation) section, pure rotation section, and pure translation section. Each of them consisted of 30 frames. The motion was generated randomly with maximum translation ± 0.01 m along x, y, and z-axes and maximum rotation $\pm 1^\circ$ about x, y,

Table 4.1: List of approaches tested in the synthetic experiment of pose tracking of two pairs of stereo cameras.

Name	Description
Our EKF-4 approach	Our EKF-based approach for pose tracking of two pairs of stereo cameras described in section 4.3.4.
Our UKF-4 approach	Our UKF-based approach for pose tracking of two pairs of stereo cameras described in section 4.3.5.
Our EKF-2 approach	Our EKF-based approach for pose tracking of a pair of stereo cameras described in section 3.3.4.
Our UKF-2 approach	Our UKF-based approach for pose tracking of a pair of stereo cameras described in section 3.3.5.
Yu's approach [38]	The approach proposed by Yu et. al. [38]. It makes use of the extended Kalman filter with the trifocal tensor constraints and the approximated twist motion model.

and z-axes per frame. Zero-mean Gaussian noises of translation 0.001 m and rotation 0.1° standard deviation were added to the motion parameters to simulate the non-smoothness in the real world. 50 independent tests were carried out in the experiment.

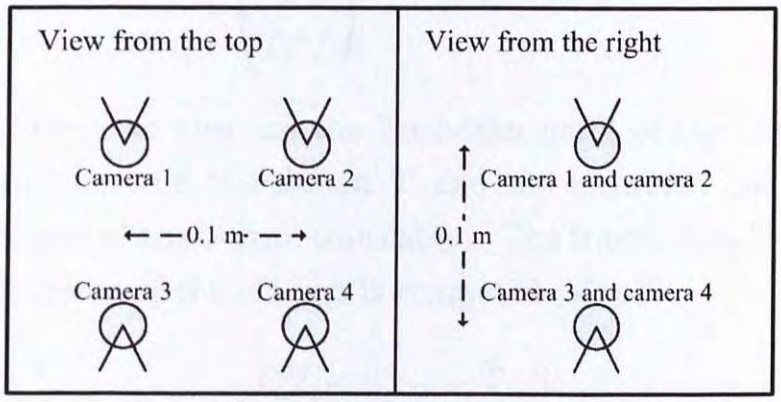

Figure 4.6: Setting 1 of the two pairs of stereo cameras in the synthetic experiment.

To compare the results, we extracted roll angle (rotation about x-axis), pitch angle (rotation about y-axis), and yaw angle (rotation about z-axis) from the recovered rotation $\hat{\mathbf{R}}$ to compare with those extracted from the true rotation $\mathbf{R}$. The rotational errors in the tth frame of the ith test are computed using

$$\begin{aligned} Roll_{t,i}^{err} &= |Roll_{\hat{\mathbf{R}}_{t,i}} - Roll_{\mathbf{R}_{t,i}}| \\ Pitch_{t,i}^{err} &= |Pitch_{\hat{\mathbf{R}}_{t,i}} - Pitch_{\mathbf{R}_{t,i}}| \\ Yaw_{t,i}^{err} &= |Yaw_{\hat{\mathbf{R}}_{t,i}} - Yaw_{\mathbf{R}_{t,i}}| \end{aligned} \tag{4.26}$$

For the translation, we extracted translations along x-axis, y-axis, and z-axis from the recovered translation $\hat{\mathbf{T}}$ to compare with those extracted from the true translation $\mathbf{T}$. The trans-

lational errors in the tth frame of the ith test are computed using

$$\begin{pmatrix} T_{x_{t,i}}^{err} \\ T_{y_{t,i}}^{err} \\ T_{z_{t,i}}^{err} \end{pmatrix} = |\mathbf{T}_{t,i} - \hat{\mathbf{T}}_{t,i}| \tag{4.27}$$

In addition, we also use the Euclidean norm of the difference between the true translation $\mathbf{T}$ and the recovered one $\hat{\mathbf{T}}$ for comparison of the overall translation. The translational error in the tth frame of the ith test is computed using

$$T_{t,i}^{err} = ||\mathbf{T}_{t,i} - \hat{\mathbf{T}}_{t,i}|| \tag{4.28}$$

Table 4.2 shows the average pose errors per frame, average number of correspondences per frame, and average processing time per frame of 50 independent tests.

The first four rows of table 4.2 compare these approaches in terms of three rotation angle errors and root-mean-square translation error. Our EKF-4 approach and UKF-4 approach could recover more accurate poses. The improvement was quite significant. As shown in the fifth row, the average number of available correspondences in our approach was roughly doubled. It is reasonable as there is one more pair of stereo cameras. The characteristic enables our approach to recover more accurate pose information. Although the processing time of our EKF-4 approach was longer than that of the approach involving only one pair of stereo cameras as shown in the sixth row, it is believed that the processing time is still short enough for real time

Table 4.2: Results of the synthetic experiment of pose tracking of two pairs of stereo cameras.

Name	Our EKF-4 approach	Our UKF-4 approach	Yu's approach [38]	Our EKF-2 approach	Our UKF-2 approach
Average errors of roll angle per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} Roll_{t,i}^{err}}{T\times N}$ (degree)	0.062°	0.062°	0.119°	0.113°	0.112°
Average errors of pitch angle per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} Pitch_{t,i}^{err}}{T\times N}$ (degree)	0.071°	0.071°	0.101°	0.096°	0.096°
Average errors of yaw angle per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} Yaw_{t,i}^{err}}{T\times N}$ (degree)	0.284°	0.284°	0.319°	0.300°	0.300°
Average errors of translation per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} T_{t,i}^{err}}{T\times N}$ (meter)	0.0079m	0.0080m	0.0123m	0.0121m	0.0121m
Average number of available correspondences per frame	45.74	45.74	25.52	25.52	25.52
Average processing time per frame (second)	0.067s	0.219s	0.026s	0.029s	0.115s
N is the number of tests which is 50. T is the number of frames which is 90.					
All the algorithms were implemented in Matlab 7.4 and run on a computer with Pentium D 3.0 GHz CPU and 1 GB RAM.					

processing.

Both the EKF-4 approach and UKF-4 approach achieved similar accuracy. It means that it is suitable to assume that the system is locally linear. Similar to the comparison between our EKF-2 and UKF-2 approaches, the processing time of the UKF-4 approach was relatively longer than that of the EKF-4 approach. It is because computing the function $\mathbf{H}_t^*(\mathbf{X}_{t|t-1})$ in the unscented transform to propagate the mean and covariance is a time consuming task in the UKF when compared with the Jacobain matrix calculation in the EKF. But the comparison may not be fair. The UKF-4 approach contains many looping statements which Matlab is weak for while the processing time of the EKF-4 approach depends on how the Jacobian matrix equation is simplified in the implementation. However, the UKF-4 approach can still work in real-time when there are less than 50 correspondences. Although both of them achieved similar accuracy, there is an issue about the implementation. It is difficult to implement the EKF-4 approach as the Jacobian matrix equation must be derived. The derivation is difficult as the measurement model is quite complex. It becomes a trade-off to choose between the EKF and the UKF. If the processing time is crucial, the EKF should be used. If the ease of implementation is crucial, the UKF should be used.

In addition, we investigated the performance of different orientations of the two pairs of the stereo cameras. We have performed another synthetic experiment to study five settings of the multiple camera system which are illustrated in figure 4.6, figure 4.7, figure 4.8, figure 4.9, and figure 4.10. The setting 1

was the same as that in the previous experiment. In setting 2, the two pairs of stereo cameras were placed back-to-back. But one stereo pair was placed horizontally while another pair was placed vertically. The facing directions of all the two stereo pairs in setting 3, setting 4, and setting 5 were perpendicular with each other in different orientations. All the configurations in the synthetic experiment were the same as the previous experiment except the setting of the multiple camera system. Table 4.3, table 4.4, table 4.5, table 4.6, and table 4.7 show the results of our EKF-4 and UKF-4 approaches for setting 1, setting 2, setting 3, setting 4, and setting 5 of the multiple camera system respectively.

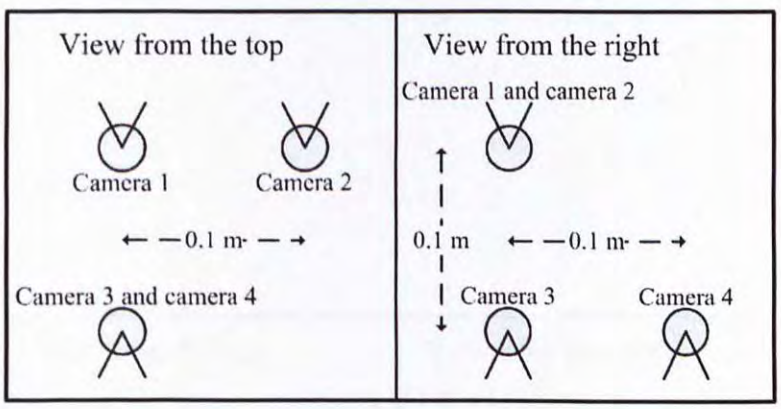

Figure 4.7: Setting 2 of the two pairs of stereo cameras in the synthetic experiment.

By comparing table 4.3, table 4.4, table 4.5, table 4.6, and table 4.7, we can see that setting 3, setting 4, and setting 5 are obviously better than setting 1 and setting 2 while the numbers of available correspondences were more or less the same. The major difference is that the facing directions of the two pairs of stereo cameras are perpendicular with each other in setting 3,

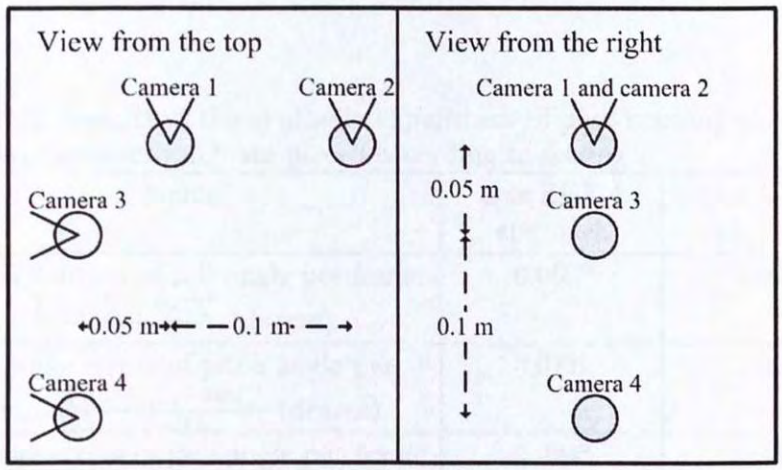

Figure 4.8: Setting 3 of the two pairs of stereo cameras in the synthetic experiment.

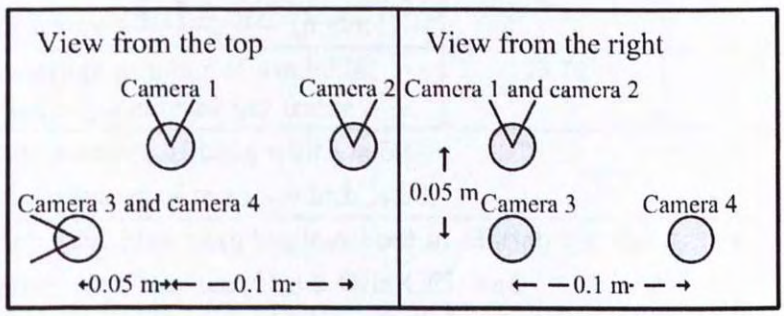

Figure 4.9: Setting 4 of the two pairs of stereo cameras in the synthetic experiment.

Table 4.3: Results of the synthetic experiment of pose tracking of two pairs of stereo cameras which are placed according to setting 1.

Name	Our EKF-4 approach	Our UKF-4 approach
Average errors of roll angle per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} Roll_{t,i}^{err}}{T \times N}$ (degree)	0.062°	0.062°
Average errors of pitch angle per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} Pitch_{t,i}^{err}}{T \times N}$ (degree)	0.071°	0.071°
Average errors of yaw angle per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} Yaw_{t,i}^{err}}{T \times N}$ (degree)	0.284°	0.284°
Average errors of x-translation per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} T_{x_{t,i}}^{err}}{T \times N}$ (meter)	0.0021m	0.0021m
Average errors of y-translation per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} T_{y_{t,i}}^{err}}{T \times N}$ (meter)	0.0021m	0.0021m
Average errors of z-translation per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} T_{z_{t,i}}^{err}}{T \times N}$ (meter)	0.0068m	0.0068m
Average errors of overall translation per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} T_{t,i}^{err}}{T \times N}$ (meter)	0.0079m	0.0080m
Average number of available correspondences per frame	45.74	45.74
N is the number of tests which is 50. T is the number of frames which is 90.		
All the algorithms were implemented in Matlab 7.4 and run on a computer with Pentium D 3.0 GHz CPU and 1 GB RAM.		

Table 4.4: Results of the synthetic experiment of pose tracking of two pairs of stereo cameras which are placed according to setting 2.

Name	Our EKF-4 approach	Our UKF-4 approach
Average errors of roll angle per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} Roll_{t,i}^{err}}{T\times N}$ (degree)	0.054°	0.054°
Average errors of pitch angle per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} Pitch_{t,i}^{err}}{T\times N}$ (degree)	0.055°	0.055°
Average errors of yaw angle per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} Yaw_{t,i}^{err}}{T\times N}$ (degree)	0.231°	0.231°
Average errors of x-translation per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} T_{x_{t,i}}^{err}}{T\times N}$ (meter)	0.0019m	0.0019m
Average errors of y-translation per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} T_{y_{t,i}}^{err}}{T\times N}$ (meter)	0.0022m	0.0022m
Average errors of z-translation per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} T_{z_{t,i}}^{err}}{T\times N}$ (meter)	0.0064m	0.0066m
Average errors of overall translation per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} T_{t,i}^{err}}{T\times N}$ (meter)	0.0075m	0.0076m
Average number of available correspondences per frame	44.04	44.04
N is the number of tests which is 50. T is the number of frames which is 90.		
All the algorithms were implemented in Matlab 7.4 and run on a computer with Pentium D 3.0 GHz CPU and 1 GB RAM.		

Table 4.5: Results of the synthetic experiment of pose tracking of two pairs of stereo cameras which are placed according to setting 3.

Name	Our EKF-4 approach	Our UKF-4 approach
Average errors of roll angle per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} Roll_{t,i}^{err}}{T\times N}$ (degree)	0.065°	0.065°
Average errors of pitch angle per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} Pitch_{t,i}^{err}}{T\times N}$ (degree)	0.053°	0.053°
Average errors of yaw angle per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} Yaw_{t,i}^{err}}{T\times N}$ (degree)	0.075°	0.076°
Average errors of x-translation per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} Tx_{t,i}^{err}}{T\times N}$ (meter)	0.0025m	0.0025m
Average errors of y-translation per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} Ty_{t,i}^{err}}{T\times N}$ (meter)	0.0017m	0.0017m
Average errors of z-translation per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} Tz_{t,i}^{err}}{T\times N}$ (meter)	0.0029m	0.0028m
Average errors of overall translation per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} T_{t,i}^{err}}{T\times N}$ (meter)	0.0049m	0.0049m
Average number of available correspondences per frame	39.05	39.05
N is the number of tests which is 50. T is the number of frames which is 90.		
All the algorithms were implemented in Matlab 7.4 and run on a computer with Pentium D 3.0 GHz CPU and 1 GB RAM.		

Table 4.6: Results of the synthetic experiment of pose tracking of two pairs of stereo cameras which are placed according to setting 4.

Name	Our EKF-4 approach	Our UKF-4 approach
Average errors of roll angle per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} Roll_{t,i}^{err}}{T\times N}$ (degree)	0.065°	0.065°
Average errors of pitch angle per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} Pitch_{t,i}^{err}}{T\times N}$ (degree)	0.050°	0.050°
Average errors of yaw angle per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} Yaw_{t,i}^{err}}{T\times N}$ (degree)	0.078°	0.078°
Average errors of x-translation per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} T_{x_{t,i}}^{err}}{T\times N}$ (meter)	0.0023m	0.0023m
Average errors of y-translation per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} T_{y_{t,i}}^{err}}{T\times N}$ (meter)	0.0017m	0.0017m
Average errors of z-translation per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} T_{z_{t,i}}^{err}}{T\times N}$ (meter)	0.0025m	0.0024m
Average errors of overall translation per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} T_{t,i}^{err}}{T\times N}$ (meter)	0.0043m	0.0043m
Average number of available correspondences per frame	38.19	38.19
N is the number of tests which is 50. T is the number of frames which is 90.		
All the algorithms were implemented in Matlab 7.4 and run on a computer with Pentium D 3.0 GHz CPU and 1 GB RAM.		

Table 4.7: Results of the synthetic experiment of pose tracking of two pairs of stereo cameras which are placed according to setting 5.

Name	Our EKF-4 approach	Our UKF-4 approach
Average errors of roll angle per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} Roll_{t,i}^{err}}{T\times N}$ (degree)	0.052°	0.052°
Average errors of pitch angle per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} Pitch_{t,i}^{err}}{T\times N}$ (degree)	0.052°	0.052°
Average errors of yaw angle per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} Yaw_{t,i}^{err}}{T\times N}$ (degree)	0.068°	0.068°
Average errors of x-translation per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} Tx_{t,i}^{err}}{T\times N}$ (meter)	0.0019m	0.0019m
Average errors of y-translation per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} Ty_{t,i}^{err}}{T\times N}$ (meter)	0.0024m	0.0024m
Average errors of z-translation per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} Tz_{t,i}^{err}}{T\times N}$ (meter)	0.0027m	0.0027m
Average errors of overall translation per frame $\frac{\sum_{t=1}^{T}\sum_{i=1}^{N} T_{t,i}^{err}}{T\times N}$ (meter)	0.0048m	0.0048m
Average number of available correspondences per frame	42.68	42.68
N is the number of tests which is 50. T is the number of frames which is 90.		
All the algorithms were implemented in Matlab 7.4 and run on a computer with Pentium D 3.0 GHz CPU and 1 GB RAM.		

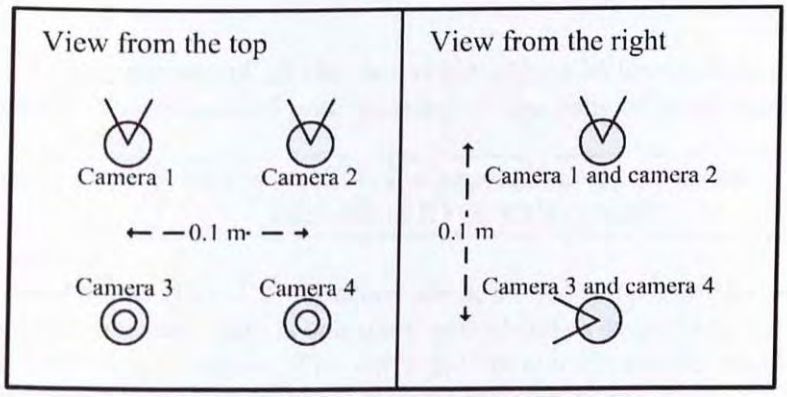

Figure 4.10: Setting 5 of the two pairs of stereo cameras in the synthetic experiment.

setting 4, and setting 5 while the two pairs are just placed back-to-back in setting 1 and setting 2. The errors of pitch angle and translation along z-axis were mostly affected since the displacement of a camera in z-axis and rotation of a camera about z-axis are not sensitive in the image. When the facing directions of the two pairs of stereo cameras are perpendicular to each other, each pair compensates for each other. As a result, setting 3, setting 4, and setting 5 should have better results. Differences between setting 1 and setting 2 and differences among setting 3 to 5 are not compared as there are no big differences of those pose errors with slightly different numbers of available features.

Table 4.8, table 4.9, and table 4.10 summarize the performances of all the tested algorithms in the synthetic experiment.

4.4.2 Real experiments

Experiment using real image sequences taken by the multiple camera system has been performed. The sequences with ground

Table 4.8: Comparison of all the tested algorithms in terms of accuracies in the synthetic experiment of pose tracking of two pairs of stereo cameras.

Accuracy:	EKF-4 $\approx$ UKF-4 > approaches of one stereo cameras (EKF-2, UKF-2, etc)
Explanation: Our EKF-4 and UKF-4 approaches are more accurate than the existing approaches because there is one more pair of stereo cameras in our EKF-4 and UKF-4 approaches. The extra pair of stereo cameras enables the system to get more features to compute the camera motion. Our EKF-4 and UKF-4 approaches achieve similar accuracies. The UKF, achieves the second order accuracy in the Taylor series, uses the unscented transform to propagate the mean and covariance of the system while the EKF, achieves the first order accuracy, propagates the mean and covariance by assuming local linearization. However, our experiment shows that both of them have similar accuracies for our problem. We believe that it is because the higher orders are not significant in our system. As a result, the EKF is already enough for our system.	

Table 4.9: Comparison of all the tested algorithms in terms of efficiencies in the synthetic experiment of pose tracking of two pairs of stereo cameras.

Efficiency:	approaches of one stereo cameras (EKF-2, UKF-2, etc) > EKF-4 > UKF-4
Explanation: Our EKF-4 approach is slower than our EKF-2 approach because the measurement function in our EKF-4 approach is more complex than that in the EKF-2 approach. Our UKF-4 approach is slower than our EKF-4 approach because more time is spent on propagating the mean and covariance of the system in the UKF. Only the Jacobian calculation is required in the EKF while the unscented transform is needed in the UKF. The unscented transform needs to the function $\mathbf{h}_t^*(\mathbf{X}_{t\|t-1})$, which is a time consuming task.	

Table 4.10: Comparison of different facing directions of the two stereo pairs in terms of accuracies in the synthetic experiment of pose tracking of two pairs of stereo cameras.

Orientation:	Perpendicular facing directions > Parallel facing directions
Explanation: The displacement of a camera along z-axis and the rotation of a camera about z-axis are not sensitive in the images. When the facing directions of the two pairs of stereo cameras are perpendicular to each other, each pair compensates for each other. As a result, displacements along all axes and rotations about all axes are sensitive in the images.	

truth data were used to evaluate the performances of our EKF-4 approach and UKF-4 approach in pose tracking of two pairs of stereo cameras. The objective of the experiment is to show that the proposed algorithm work accurately in the real world.

In the real experiment, four web cameras with resolution 320×240 shown in figure 4.11 were mounted on top of a robot shown in figure 3.10. The robot was driven by two servo motors that were attached to the wheels on the left and right. A personal computer sent control signals to control its movements. To change the direction, two wheels were made to move at different motions. For instance, moving left motor forward and right motor backward could make the robot turn right at a certain degree. Images taken by the cameras were transferred to the personal computer via Universal Serial Bus (USB). Given the diameters of the wheels, distance between them and robot displacement per motor step, we could compute the actual orientation and position of the robot and thus the multiple camera

system (ground truth).

Figure 4.11: The two pairs of stereo cameras mounted on the robot in the real experiment.

To compare the results, we extracted roll, pitch and yaw angles from the recovered rotation to make a comparison with those angles from ground truth. For the translation, we extracted translations along x-axis, y-axis and z-axis from the recovered translation to make a comparison with those true translations from ground truth.

Three sets of image sequences taken by the multiple camera system were used in the real experiment. The first set consisted of 200 frames. The images from the 4 cameras at the first frame of the first image sequences are shown in figure 4.12. The motion of the multiple camera system consisted of both translation and rotation. The number of the correspondences available in the stereo image pair by camera 1 and camera 2 was between 5 and 22 while that in the stereo image pair by camera 3 and camera 4 was between 21 and 59. The recovered poses are shown in figure

4.16. Table 4.11 shows the timings of the experiment.

The second set consisted of 150 frames. The images from the 4 cameras at the first frame of the second image sequences are shown in figure 4.13. Similar to the first set, the motion of the system consisted of both rotation and translation. However, the motion and scene were different from those in the first set. The number of the correspondences available in the stereo pair by camera 1 and camera 2 was between 1 and 20 while that in the stereo pair by camera 3 and camera 4 was between 12 and 38. The experimental result is shown in figure 4.17. Table 4.12 shows the timings of the experiment.

The third set consisted of 80 frames. The images from the 4 cameras at the first frame of the third image sequences are shown in figure 4.14. Different from the first two set, something was blocking the stereo pair consisting of camera 1 and camera 2 from the 41st to 43rd frame and thus no correspondences were found in the stereo image pair in this period. Figure 4.15 shows the images from camera 1 and camera 2 at the 42nd frame. The number of the correspondences available in the stereo pair by camera 1 and camera 2 was between 0 and 61 while that in the stereo pair by camera 3 and camera 4 was between 5 and 48. The experimental result is shown in figure 4.18. Table 4.13 shows the timings of the experiment.

In figure 4.16, figure 4.17 and figure 4.18, lines with ($\bigcirc$) show the ground truth. Lines with ($+$) show the recovered pose by our EKF-4 approach while lines with ($\times$) show the recovered pose by our UKF-4 approach. The ability of our approaches was shown by comparing the ground truth data and the recovered pose

information by our approaches. We can see that our approaches can recover the orientation and location of the multiple camera system accurately. Our approach can recover the pose even when one stereo pair was blocked as shown in figure 4.15 in the third testing sequence.

In table 4.11, table 4.12, and table 4.13, the first row shows the times used per frame in feature detection and tracking of each camera. The second row shows the times used per frame in stereo correspondence matching of each stereo pair. The third row shows the times used per frame in pose tracking of each tested algorithm.

Table 4.14 summarizes the results of the synthetic and real experiments.

Table 4.11: Timings of the real experiment of pose tracking of two pairs of stereo cameras using the first stereo image sequences.

	Camera 1	Camera 2	Camera 3	Camera 4
Feature detection and tracking (second per frame)	0.171s	0.166s	0.164s	0.167s
Stereo correspondence matching (second per frame)	0.169s		0.264s	
Camera pose tracking (second per frame)	EKF-4: 0.058s UKF-4: 0.201s			
All the algorithms were implemented in Matlab 7.4 and run on a computer with Pentium D 3.0 GHz CPU and 1 GB RAM.				

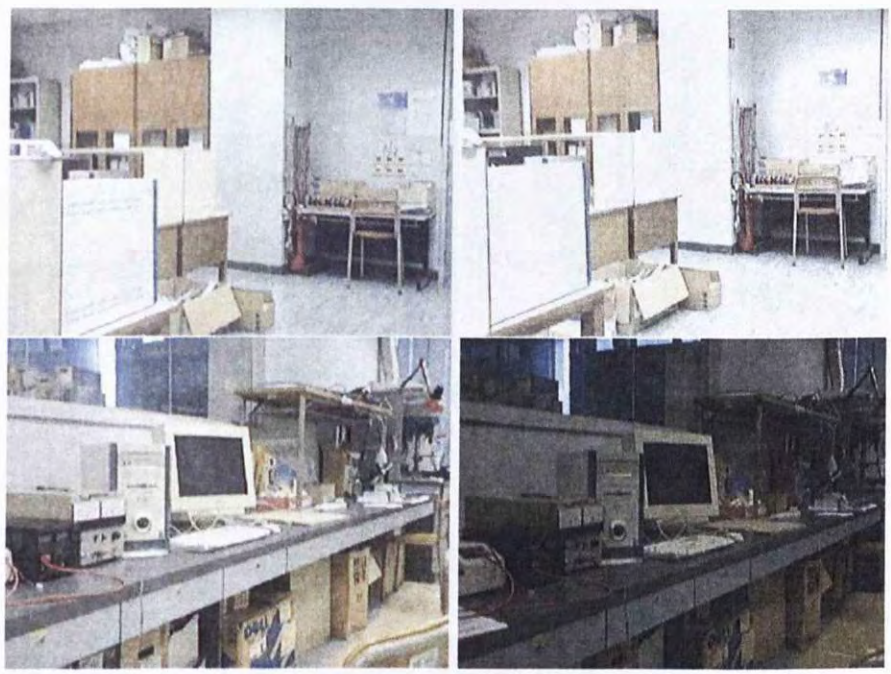

Figure 4.12: The images at the first frame of the first image sequences in the real experiment of pose tracking of two pairs of stereo cameras. (Top left) Image from camera 1. (Top right) Image from camera 2. (Bottom left) Image from camera 3. (Bottom Right) Image from camera 4.

Figure 4.13: The images at the first frame of the second image sequences in the real experiment of pose tracking of two pairs of stereo cameras. (Top left) Image from camera 1. (Top right) Image from camera 2. (Bottom left) Image from camera 3. (Bottom Right) Image from camera 4.

Figure 4.14: The images at the first frame of the third image sequences in the real experiment of pose tracking of two pairs of stereo cameras. (Top left) Image from camera 1. (Top right) Image from camera 2. (Bottom left) Image from camera 3. (Bottom Right) Image from camera 4.

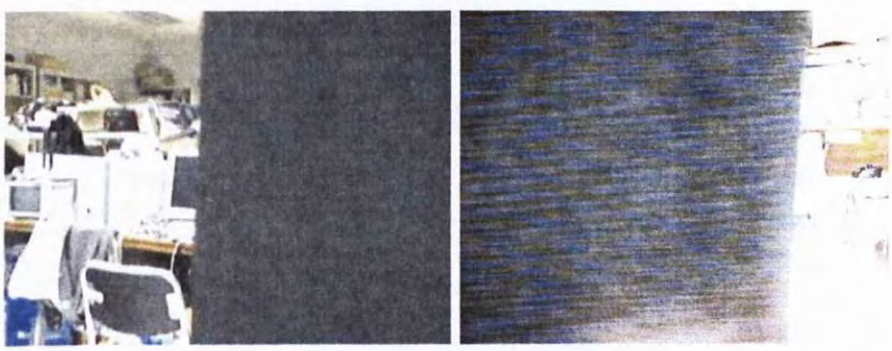

Figure 4.15: The images from camera 1 and camera 2 at the 42nd frame of the third image sequences in the real experiment of pose tracking of two pairs of stereo cameras. (Left) Image from camera 1. (Right) Image from camera 2.

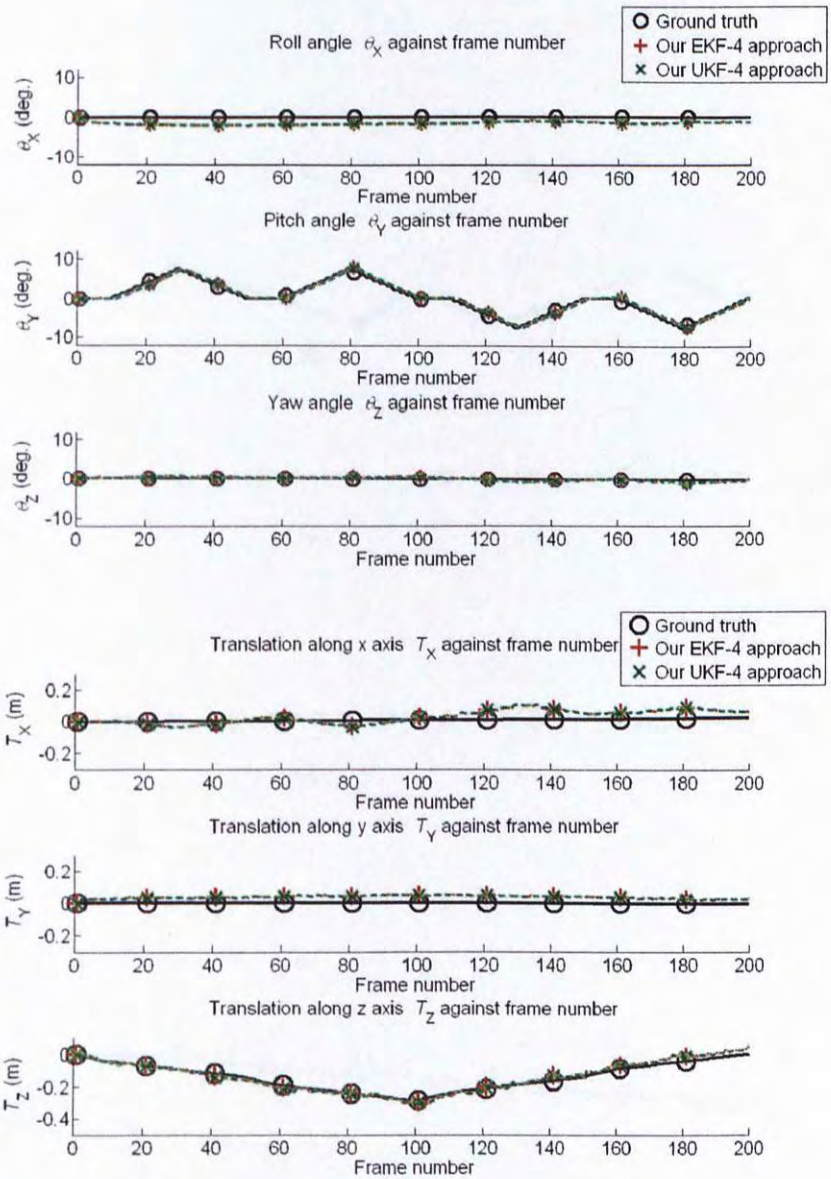

Figure 4.16: Result of the real experiment of pose tracking of two pairs of stereo cameras using the first image sequences. (Top) Rotation. (Bottom) Translation.

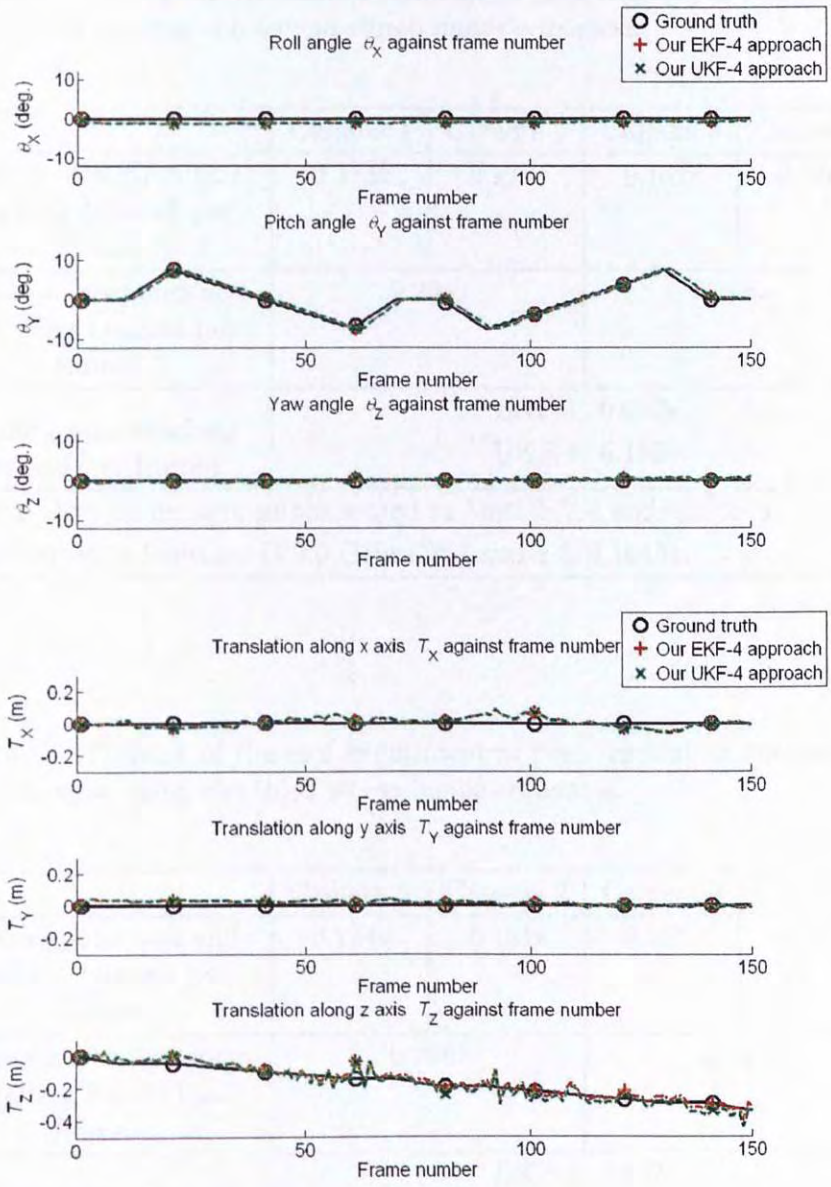

Figure 4.17: Result of the real experiment of pose tracking of two pairs of stereo cameras using the second image sequences. (Top) Rotation. (Bottom) Translation.

Table 4.12: Timings of the real experiment of pose tracking of two pairs of stereo cameras using the second stereo image sequences.

	Camera 1	Camera 2	Camera 3	Camera 4
Feature detection and tracking (second per frame)	0.171s	0.172s	0.162s	0.165s
Stereo correspondence matching (second per frame)	0.225s		0.225s	
Camera pose tracking (second per frame)	EKF-4: 0.037s UKF-4: 0.133s			
All the algorithms were implemented in Matlab 7.4 and run on a computer with Pentium D 3.0 GHz CPU and 1 GB RAM.				

Table 4.13: Timings of the real experiment of pose tracking of two pairs of stereo cameras using the third stereo image sequences.

	Camera 1	Camera 2	Camera 3	Camera 4
Feature detection and tracking (second per frame)	0.184s	0.181s	0.181s	0.180s
Stereo correspondence matching (second per frame)	0.205s		0.194s	
Camera pose tracking (second per frame)	EKF-4: 0.037s UKF-4: 0.148s			
All the algorithms were implemented in Matlab 7.4 and run on a computer with Pentium D 3.0 GHz CPU and 1 GB RAM.				

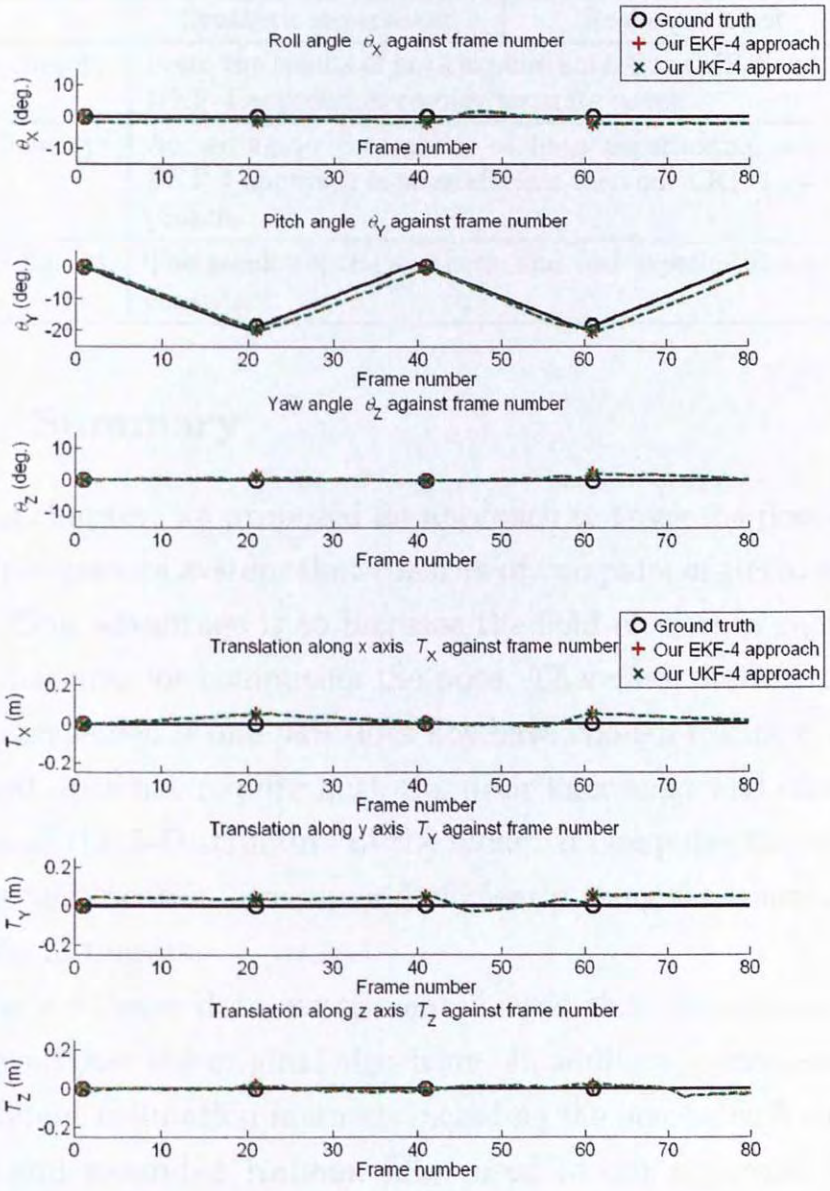

Figure 4.18: Result of the real experiment of pose tracking of two pairs of stereo cameras using the third image sequences. (Top) Rotation. (Bottom) Translation.

Table 4.14: Summary of the results of the synthetic and real experiments.

	Synthetic experiment	Real experiment
Accuracy:	From the results of both experiments, Our EKF-4 and UKF-4 approaches recover accurate poses.	
Efficiency:	According to the results of both experiments, our EKF-4 approach is more efficient than our UKF-4 approach.	
Conclusion:	The results of the synthetic and real experiment are consistent.	

4.5 Summary

In this chapter, we proposed an approach to track the pose of a multiple camera system that consists of two pairs of stereo cameras. One advantage is to increase the field of view to capture more features for computing the pose. Therefore, our approach still works even if one pair does not have enough features. Our method does not require both the prior knowledge and computation of the 3-D structure of the scene. It computes the orientation and location directly and efficiently using the constraints of trifocal tensors.

The synthetic data experiment showed that the accuracy is improved over the original algorithm. In addition, performances of different estimation methods including the unscented Kalman filter and extended Kalman filter used in our approach were compared and analyzed. The experiment demonstrated that the EKF-based approach and UKF-based approach have similar performances in terms of accuracy. In addition, effects of different orientations between the two pairs of stereo cameras were

studied. It was found that the recovered pose is more accurate when the facing directions of the two pairs of stereo cameras are perpendicular to each other.

Real image experiment showed that the recovered poses by our approach are accurate when they are compared with the ground truths. The approaches are believed to be useful for navigating robot and building augmented reality systems.

□ **End of chapter.**

Chapter 5

Conclusion

5.1 Conclusion

The thesis focuses on pose tracking of multiple camera systems based on the model-less scheme in which both the prior knowledge and explicit computation of the 3-D structure are not required. Firstly, a survey of popular algorithms for camera pose estimation was conducted.

We then proposed an algorithm based on the model-less scheme to estimate the pose of a pair of stereo cameras. In the experiment, it was found that the proposed algorithm can recover more accurate pose information than the current algorithms. We also have compared and analyzed the performances of different estimation methods including the extended Kalman filter, unscented Kalman filter, and differential evolution used in our approach throughout the experiment.

Finally, we proposed an algorithm to track the pose of a multiple camera system consisting of two pairs of stereo cameras. One advantage is to increase the field of view to capture

more features for computing the pose. Therefore, our approach still works even if one pair does not have enough features. Our method does not require both the prior knowledge and explicit computation of the 3-D structure of the scene. It computes the orientation and location efficiently and directly by employing the constraints of trifocal tensors. In the experiments, we also applied and compared different estimation methods including the extended Kalman filter and unscented Kalman filter in our approach and investigated how different orientations between the two pairs of stereo cameras affect the accuracy of the recovered pose.

5.2 Scope of Applications

We believe that our approaches and studies are useful for applications related to the camera pose estimation like building augmented reality system, robot navigation, motion sensing, and etc.

In building augment reality system, artificial objects are inserted into the films. When the camera motion is known, the inserted objects can be moved accordingly in the films.

To navigate a robot, it is essential to know the position of the robot. Our multiple camera system can be mounted on the robot to track the motion of the robot.

In motion sensing, the multiple camera system can be mounted on any mobile devices to detect their motions.

5.3 Limitations

There are some limitations for our system. The major limitation is that, similar to all the other vision based approaches using cameras, the environment for our approaches should be rich in features. For example, if the view of the camera is only a white wall, the number of features is very limited, and thus it is impossible to recover the camera motion. To estimate the pose using the constraint of trifocal tensor, there should be at least seven feature correspondences. Otherwise, our approach is not stable. In the real situation, seven feature correspondences are still not enough as there should be noise originated from the cameras. Having more feature correspondences is more resistant from noise. Depending on the amount of the noise, the acceptable number of the feature correspondences is different. But it is recommended to contain 20 - 50 feature correspondences. However, there is one advantage of our approach for the pose estimation of two pair of stereo cameras. Our approach can still work even when there are not enough features from one of the stereo pairs.

Another major limitation is that there should be no moving objects in the scene. When there is moving objects, the features associated to the moving objects cannot help to track the camera pose. Including them to estimate the pose would increase the error of the recovered pose. As a result, special treatment should be conducted to identify such features. Further investigation would be conducted in the future.

5.4 Difficulties

There were some difficulties in the implementation of the proposed approaches. Firstly, as the measurement models in our approaches are quite complex, it is difficult to compute their corresponding Jacobians in the extended Kalman filter. It is contrary to the unscented Kalman filter in which no Jacobian calculation is involved.

Another difficulty is the calibration of the cameras. Although there are some calibration methods, it is still difficult to calibrate the relationships between the cameras accurately as the methods are quite troublesome.

5.5 Future work

Different estimation methods including the extended Kalman filter and the unscented Kalman filter have been studied for our problem. Some more advanced techniques can be studied to compute the reliability of each feature from both of the two stereo pairs or the reliability of the set of features from each stereo pair. After the reliabilities of them are got, recovery of camera motion can rely on those with high reliabilities. For example, when one pair of stereo cameras is occluded, the features detected and tracked from this stereo pair is highly unreliable. So it is better to ignore the features from this stereo pair. To deal with this problem, some probabilistic methods can be employed to detect such situations and then handle them properly. Examples of the possible methods are embedding interacting

multiple model or probabilistic data association filter into the Kalman filters.

Our study of the differential evolution is restricted to one set of parameters. Therefore, comprehensive evaluation using different settings can be conducted. For example, we can study how different population sizes, different numbers of generations, different amplification factors, different crossover rates affect the accuracy of the recovered pose. Even different variants of differential evolution can also be studied to see which one is suitable for our problem. However, even the best setting is found, it is not surprising that the differential evolution may still be worse than other recursive filters like the extended Kalman filter and the unscented Kalman filter in terms of running timed as evolutionary algorithms are generally slow.

In our approaches of pose estimation of the multiple camera system, the relationships of all the cameras in the system are fixed. It is valuable to investigate a multiple camera system consisting of movable cameras. Consider that each camera in the system can be moved independently, it is interesting to investigate how each individual camera moves can benefit the whole system. For example, the cameras can be moved to obtain as many as reliable features to compute the pose.

There are no special managements of the features in the proposed approaches. When the multiple camera system moves, new features may appear and original features may disappear. It is valuable to investigate more advanced methods to handle the features. For example, consider that features which disappear in the earlier frame reappear in the latter frame, if we can

recognize these features, it is expected that more accurate pose can be recovered.

Lastly, some advanced techniques can be investigated to handle the scene having moving objects. They would make the whole system more robust and suitable for the real situation.

□ **End of chapter.**

Bibliography

[1] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Signal Process.*, 50(2):174–188, Feb 2002.

[2] A. Azarbayejani and A. P. Pentland. Recursive estimation of motion, structure, and focal length. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(6):562–575, Jun. 1995.

[3] P. Baker, C. Fermuller, Y. Aloimonos, and R. Pless. A spherical eye from multiple cameras (makes better models of the world). In *IEEE Conf. Computer Vision and Pattern Recognition*, volume 1, pages I–576–I–583, 2001.

[4] P. Baker, A. S. Ogale, and C. Fermuller. The argus eye: a new imaging system designed to facilitate robotic tasks of motion. *IEEE Robotics & Automation Magazine*, 11(4):31–38, Dec. 2004.

[5] J.-Y. Bouguet. Camera Calibration Toolbox for Matlab. Open source software. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/.

[6] T. J. Broida, S. Chandrashekhar, and R. Chellappa. Recursive 3-D motion estimation from a monocular image sequence. *IEEE Trans. Aerosp. Electron. Syst.*, 26(4):639–656, Jul. 1990.

[7] P. Cerveri, A. Pedotti, and N. A. Borghese. Combined evolution strategies for dynamic calibration of video-based measurement systems. *IEEE Trans. Evol. Comput.*, 5(3):271–282, Jun 2001.

[8] D. Chekhlov, M. Pupilli, W. Mayol-Cuevas, and A. Calway. Real-time and robust monocular SLAM using predictive multi-resolution descriptors. In *2nd Int. Symp. Visual Computing, Part II*, pages 276–285, Nov. 2006.

[9] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6):1052–1067, Jun. 2007.

[10] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[11] M. S. Grewal and A. P. Andrews. *Kalman Filtering: Theory and Practice.* Prentice Hall, 1993.

[12] M. D. Grossberg and S. K. Nayar. A general imaging model and a method for finding its parameters. In *IEEE Int. Conf. Computer Vision*, volume 2, pages 108–115, 2001.

[13] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, second edition, 2004.

[14] S. Hati and S. Sengupta. Robust camera parameter estimation using genetic algorithm. *Pattern Recogn. Lett.*, 22(3-4):289–298, 2001.

[15] Q. Ji and Y. Zhang. Camera calibration with genetic algorithms. *IEEE Trans. Syst., Man, Cybern. A*, 31(2):120–130, Mar 2001.

[16] S. Julier and J. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls, Orlando, FL*, 1997.

[17] S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, Mar 2004.

[18] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proc. American Control Conf.*, volume 3, pages 1628–1632, 21-23 Jun. 1995.

[19] S. Lee, S. Lee, and D. Kim. Recursive unscented Kalman filtering based SLAM using a large number of noisy observations. *Int. J. Control, Automation, and Systems*, 4(6):736–747, 2006.

[20] V. Lippiello, B. Siciliano, and L. Villani. Position and orientation estimation based on Kalman filtering of stereo images. In *IEEE Int. Conf. Control Applications*, pages 702–707, 2001.

[21] B. Lloyd. Computation of the Fundamental Matrix. Open source software. Available: http://www.cs.unc.edu/ blloyd/comp290-089/fmatrix/.

[22] T. Morita and T. Kanade. A sequential factorization method for recovering shape and motion from image streams. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(8):858–867, Aug 1997.

[23] M. Pupilli and A. Calway. Real-time visual SLAM with resilience to erratic motion. In *IEEE Conf. Computer Vision and Pattern Recognition*, volume 1, pages 1244–1249, 17-22 Jun. 2006.

[24] M. Pupilli and A. Calway. Real-time camera tracking using a particle filter. In *British Machine Vision Conf.*, 2005.

[25] D. C. Schuurman and D. W. Capson. Direct visual servoing using network-synchronized cameras and Kalman filter. In *IEEE Int. Conf. Robotics and Automation*, volume 4, pages 4191–4197, 2002.

[26] S. Soatto, R. Frezza, and P. Perona. Motion estimation on the essential manifold. In *European Conf. Computer Vision (Vol. II)*, pages 61–72. Springer-Verlag New York, Inc., 1994.

[27] B. Stenger, P. R. S. Mendonca, and R. Cipolla. Model-based 3D tracking of an articulated hand. In *IEEE Conf. Computer Vision and Pattern Recognition*, volume 2, pages 310–315, 2001.

[28] R. Storn and K. Price. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, Berkeley, CA, 1995.

[29] R. Storn and K. Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *J. of Global Optimization*, 11(4):341–359, 1997.

[30] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, Apr. 1991.

[31] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *Int. J. Comput. Vision*, 9(2):137–154, 1992.

[32] F. Toyama, K. Shoji, and J. Miyamichi. Model-based pose estimation using genetic algorithm. In *IEEE Conf. Computer Vision and Pattern Recognition*, volume 1, pages 198–201 vol.1, Aug 1998.

[33] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment - A modern synthesis. In *Proc. Int. Workshop Vision Algorithms (ICCV 99)*, pages 298–372. Springer-Verlag, 2000.

[34] E. A. Wan and R. van der Merwe. The unscented Kalman filter. In S. Haykin, editor, *Kalman Filtering and Neural Networks*, pages 221–280. Wiley-Interscience, 2001.

[35] Y. K. Yu, K. H. Wong, and M. M. Y. Chang. Pose estimation for augmented reality applications using genetic algorithm. *IEEE Trans. Syst., Man, Cybern. B*, 35(6):1295–1301, Dec. 2005.

[36] Y. K. Yu, K. H. Wong, and M. M. Y. Chang. Recursive three-dimensional model reconstruction based on Kalman filtering. *IEEE Trans. Syst., Man, Cybern. B*, 35(3):587–592, Jun. 2005.

[37] Y. K. Yu, K. H. Wong, M. M. Y. Chang, and S. H. Or. Recursive camera-motion estimation with the trifocal tensor. *IEEE Trans. Syst., Man, Cybern. B*, 36(5):1081–1090, Oct. 2006.

[38] Y. K. Yu, K. H. Wong, S. H. Or, and M. M. Y. Chang. Recursive recovery of position and orientation from stereo image sequences without three-dimensional structures. In *IEEE Conf. Computer Vision and Pattern Recognition*, volume 1, pages 1274–1274, 17-22 Jun. 2006.

[39] Z. Zhang and J. Zhang. Driver fatigue detection based intelligent vehicle control. In *Int. Conf. Pattern Recognition*, volume 2, pages 1262–1265, 2006.